Two phased Service Oriented Broker for replica selection in Data Grids

A thesis submitted during 2012 to University of Hyderabad in partial fulfillment of the requirements for the award of a Ph.D. Degree in Computer Science

by

Rafah M. Almuttairi

Regd. No. 06MCPC12

Department of Computer & Information Sciences
School of Mathematics & Computer / Information Sciences

University of Hyderabad
(P.O.) Central University, Gachibowli
Hyderabad - 500 046
Andhra Pradesh India

January 2012

CERTIFICATE

This is to certify that the thesis entitled "Two phased Service Oriented Broker for replica selection in Data Grids" submitted by R. M. Almuttairi bearing (Reg. No 06MCPC12) in partial fulfillment of the requirements for the award of Doctor of Philosophy in Computer Science is a bonafide work carried out by her under our supervision and guidance.

The thesis has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Dr. Rajeev Wankar
(Supervisor)
Department of Computer
&
Information Sciences,
University of Hyderabad,
Hyderabad.

Dr. Atul Negi
(Supervisor)
Department of Computer
&
Information Sciences,
University of Hyderabad,
Hyderabad.

Head of the Department

Dean of the School

Declaration

I, **R. M. Almuttairi**, hereby declare that this Dissertation entitled **"Two phased Service Oriented Broker for replica selection in Data Grids"** submitted by me under the guidance and supervision of **Dr. Rajeev Wankar and Dr. Atul Negi**, is a bonafide work. I also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date: January-2012

R. M. Almuttairi
Signature of the Student:
Regd. No.: 06MCPC12

Acknowledgements

Firstly I love to say:
"I am grateful to God for giving me the health and emotional strength to be able to finish my research."

During the course of my research program, I had the privilege to get help from many people. First and foremost, I would like to express my heart-felt gratitude to my Ph.D. supervisors Dr. Rajeev Wankar and Dr. Atul Negi. The support and encouragement I received from my supervisors has been exceptional. They advised me to be a good listener first and then a speaker. Throughout my research program, They showed me how one can be a good and effective researcher by imparting advice on all technical issues. They have always shown confidence in me. The best thing, apart from many others, that I could learn from them was the importance of perseverance. In all, both my supervisors contributed a lot in building my confidence as a good researcher.

I would like to thank the staff of DCIS for their encouragement throughout my research they all have been very kind and friendly. I would also like to thank **Indian Council for Cultural Relations** and **Iraqi Ministry of Higher Education** for their fellowships, without which it would have been very difficult to pursue this research. I am thankful to Prof. Dr. C. R. Raw for his invaluable suggestions which helped me in improving my work. I would like to express my gratitude to Prof. Dr. Arun Agarwal for ensuring that we had an excellent Grid Computing Laboratory with the latest computing facilities. I would like to thank Ms. Anupama Potluri, Dr. S. Durga Bhavani, Prof. Dr. Chakravarthy Bhagvati, Prof. Dr. S.Bapi Raju and P.S.V.S. Sai Prasad. for their valuable classes which I have not missed any.

I would thank **Prof. Roger Leslie Anderton Cottrell** from **Stanford Linear Accelerator Center (SLAC)** at **Stanford University** for providing the network monitoring reports of **CERN** of EU Data Grid.

I would like to thank who accompanied me throughout the study period, my dearest sister Yulia Sukonkina.

I am thankful to my colleagues and friends, Mushreq Mohammed, Asaad Al-Salih, Mustafa Kaiiali, Ahmad Jabas, Amer sallam, Rusydi Umar Pravin, Babo, Mini, Pavan Kumar and many others who have been a continuous source of support and have made my interaction with them memorable. Directly or indirectly, they have contributed in the completion of this work. My special thanks goes to.

I am deeply indebted to my family for their love and support. This thesis is dedicated to my Mom. She has always been confident about my abilities. She never imposed anything on me and has always inspired me to do better. It is from her that I have imbibed the never-say-die attitude in life. Inspite of many difficulties I could complete this work and the credit goes to my beloved husband. He was there whenever I had problems. I also take this opportunity to express my love and gratitude to my daughter. She was the one who had to bear the brunt of my mood swings. At the same time, I used to forget my stress when I was with her. May God bless her with eternal happiness. I am really proud to have them as my family I am also grateful to my sisters and brothers in the law Thikra & Reyath, Bushra & Duraid and Rwayda & Mahmood and my dear brother Hayder and his wife Eman for their care and support.

I would also thank my closest friends Rwayda, Nihal, Zainab, Gaida and Sawsan for their help and support. Finally I would like to thank my Dad who forgave my absence when he called me in the last days of his life while I was busy with my research.

Date: January-2012 R. M. Almuttairi

Contents

List of Figures

List of Tables

List of Algorithms

List of Publications

Most of the work presented in the thesis has been published or is under review for publication in referred International Journals and Conferences:

1. Rafah M. Almuttairi, RajeevWankar, Atul Negi, C. Raghavendra Rao., **Intelligent Replica Selection Strategy for Data Grid**, the 2010 International Conference on Grid Computing & Applications (GCA 2010), pp. 95-101, **IEEE Computer Society, July 2010**, ISBN 1-60132-144-9, 2010, Las Vegas, Nevada, USA [AWNR10a].

2. Rafah M. Almuttairi, Rajeev Wankar, Atul Negi, C. Raghavendra Rao., **Rough Set Clustering approach to replica selection in Data Grids (RSCDG)**, 10th International Conference on Intelligent Systems Design and Applications (ISDA'2010), pp. 1195-1200, **IEEE Computer Society, December 2010**, Cairo, Egypt [AWNR10c].

3. Rafah M. Almuttairi, Rajeev Wankar, Atul Negi, C. Raghavendra Rao., **Smart Replica Selection for Data Grids Using Rough Set Approximations (RSDG)**, the 2010 Computational Intelligence and Communication Networks (CICN), pp. 466-471, **IEEE Computer Society**, ISBN: 978-1-4244-8653-3, November 2010 , Bhopal [AWNR10d]

4. Rafah M. Almuttairi, Rajeev Wankar, Atul Negi, C. Raghavendra Rao., **New replica selection technique for binding replica sites in Data Grids**, the 2010 EPC-IQ, pp. 187-194, **IEEE Computer Society, November 2010**, Basrah, Iraq [AWN+10a].

5. Rafah M. Almuttairi, Rajeev Wankar, Atul Negi, C. Raghavendra Rao., **Replica Selection in Data Grids Using Preconditioning of Decision Attributes by K-means Clustering (K-RSDG)**, the 2010 Second Vaagdevi International Conference on Information Technology for Real World Problems (VCON2010), pp.18-23, **IEEE Computer Society, December 2010**, India [AWNR10b].

6. Praveen Ganghishetti, Rajeev Wankar, Rafah M. Almuttairi, C. Raghavendra Rao., **Rough Set Based Quality of Service Design for Service Provisioning in Clouds** , 6th International Conference on Rough Sets and Knowledge Technology (RSKT 2011) pp. 268-273, Lecture Notes in Computer Science 6954, **Springer, 2011**, ISBN 978-3-642-24424-7, October 2011, Banff, Canada [GWAR11].

7. Rafah M. Almuttairi, Rajeev Wankar, Atul Negi, C. Raghavendra Rao., **Enhanced Data Replication Broker**, 5th Multi-Disciplinary International Workshop on Artificial Intelligence (5th MIWAI-2011), pp. 286-297, LNAI 7080, **Springer, December 2011** , Hyderabad [AWNR11].

8. Rafah M. Almuttairi, Rajeev Wankar, Atul Negi, C. Raghavendra Rao., Arun Agrawal, Rajkummar Buyya, **A Two Phased Service Oriented Broker for Replica Selection in Data Grids (2SOB)**, Journal of Future Generation Computer Systems. (communicated)

Whenever results of any of these works are reported, proper references are made in the body of the thesis.

Abstract

Scientific research such as the experiments of LHC (Large Hadron Collider) generate data at a tremendous speed and volume. There is thus an urgent requirement of Data Grids that manage, distribute and access large sets of raw and processed data efficiently and effectively across the globe. A Data Grid is an architecture for a distributed data management system which uses Data Replication as one of its key components. Replication enhances data access and reliability by distributing copies of data at different locations. In grid computing, a resource broker is an agent that takes decisions to acquire grid services and resources for higher level components. Data optimization is one of the key functions a resource broker in Data Grids must perform. It needs a process that fetches data from replicas over a network in order to fulfill its design goals of cost, performance etc. Here we propose a novel approach to replica selection methods in the resource broker. The proposed architecture is compared with other policies and shown to be better than others by using standard statistical methods. The research is about investigating, selecting, modifying, and experimenting with some non-conventional replica selection approaches. Factors such as provider reliability, transfer speeds, link reliability and service costs have also been considered. The proposed broker approach is called as a A Two phased Service Oriented Broker for replica selection (2SOB). 2SOB's main features are: ***Scalability***, ***Reliability***, ***Stability***, ***Uncertainty***, ***Availability***, ***Accuracy***, ***Efficiency***, and ***Easy for deployment***.

2SOB consists of two phases; the first is a Coarse-grain phase, basically used for discovering replica sites, that have low latency (uncongested) network links, and distinguishing them from other replicas having high latency (congested network links).

Procedurally, this has been done using the association rules concept from Data Mining approach. The second phase is known as the Fine-grain phase used for sifting the plausible set of associated providers that match closely the user/ application requirements. Grey-based Rough Set Theory is used in the second phase to deal with two types of attributes, exact values and linguistic variables. Both phases have accordingly been designed, simulated, coded, and then validated on real data from EU Data Grid. The first phase has been applied on the real network data of (February, 2011) from CERN. Experimentation are carried out and results are compared with some other contemporary selection methods of different brokers, and it has been shown that our methods outperformed other previously proposed methods in both phases. Using the proposed broker, 2SOB, the selection has been enhanced in two ways, first is the speed of fetching data where our improvement is between (21% - 52%), and the second is the accuracy of selection where the improvement is between(25% - 37%). The second phase has further been modified to work as a selection strategy for a resource broker that does Storage Management in Cloud. The upgraded strategy has been applied on a model with synthetic data, based upon the storage application in the cloud environment of Amazon.

Chapter I

Introduction

The work presented within this thesis is in the larger context of Grid Computing and more specifically with reference to Data Grids. Throughout the years, many commercial and non-commercial institutes produced implementations of some kind of a distributed system, calling it “a Grid”. It’s fair to say that the term “Grid” now assumes different meanings according to the different implementers. We don’t deal with any particular implementation, but we start with the concept of what is a Grid and its purpose.

Grid systems can be placed into categories as shown in Figure 1.1. That is *Computational Grids*, *Data Grids* and *Service Grids* [KBM02]. The Computational Grid category denotes systems that have higher aggregate computational capacity available for single applications than the capacity of any constituent machine in the system. These systems can be further subdivided into two more categories: a distributed supercomputing Grid that executes the application in parallel on multiple machines to reduce the completion time of a job. The Data Grid category is for systems that provide an infrastructure for synthesizing new information from data repositories such as digital libraries or data warehouses that are distributed in a wide area network. Computational Grids also need to provide data services but the major difference between a Data Grid and a Computational Grid is the specialized infrastructure provided to applications for storage management and data access. Data Grid technologies are one of the major topics of this thesis, so a more in-deep view into Data Grids can be found in the next section.

An increasing number of scientific applications ranging from high-energy physics to computational genomics require access to large amounts of data (currently terabytes and soon petabytes) with varied quality of service (QoS) requirements. This

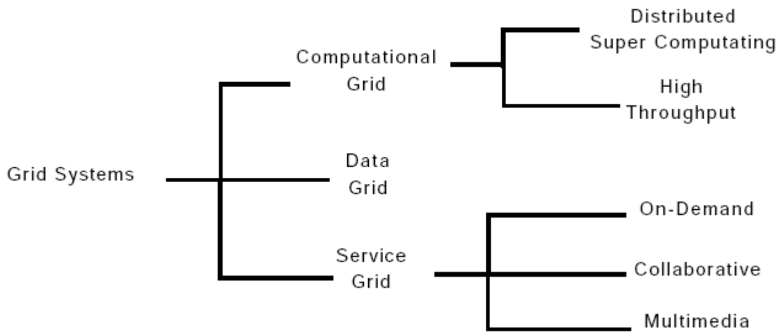

Figure 1.1: A Grid System taxonomy [VBR06]

diverse demand has contributed to the proliferation of storage system capabilities, thus making storage devices an integral part of the Grid environment and thereby constituting the Data Grid.

Grid applications have now a growing importance in the vast *e-Science* world. Generally, these applications involve the production of large datasets (viz. data collections) from simulations or from large scale experiments such as *Large Hadron Collider* (LHC) for example. Datasets are typically stored on mass storage systems, and must be accessed by users at different locations. Users may create local copies or replicas of the datasets to reduce latencies involved in wide-area data transfers. A replica may be a complete or a partial copy of the original dataset [KBM02].

For the purpose of protecting the generated data and reduce the latency of accessing data, the data should be stored at different locations. That is called replication of data. On the other hand, the data is requested for, and needs to be accessed, from a number of analysts and researchers from various parts of the world. In such a case there is a need for the replica selection process. The selection process is used to determine the best replica (file) provider. The requirements of the systems described above can be met using middleware in Data Grids.

Data Grids are distributed systems which contain a variety of data resources. Different levels of virtualizations over these data resources could be provided and resources offered as services. A Data Grid generally is also seen as a service oriented

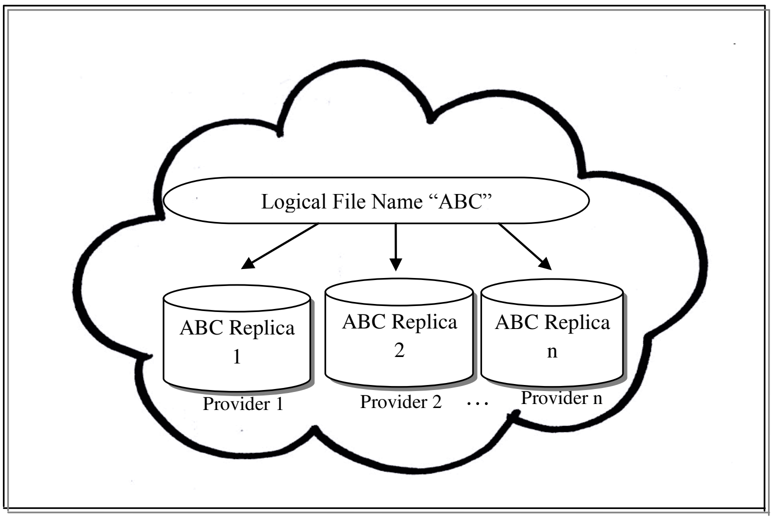

Figure 1.2: Replicas of ABC file

platform where all kind of resources (files) are treated as services for users. The main aim of data grid architecture is to provide a reliable service with an easy and scalable access to all those who search for services on the Internet. These services are designed to support data intensive applications that require access to the large amount of data (terabytes or even petabytes) with varied quality of service requirements. Data grid infrastructure provides the core services of data transport, replica cataloging, network monitoring services. These can be used to build various higher level services. One such service is the Data Grid Broker Service as shown in Figure 1.3.

In Data Grids, Replica Management System refers to the process of *selection* and *replication* of file/part of file in Data Grids as shown in Figure 1.2. In our work we deal with only selection problem which is a type of *Replica Optimization Service* (ROS). In this type of ROS we need to answer two questions:

1. From whom (replica providers) to get a file

2. How much time it takes to get this file

Often these questions are answered with respect to the QoS parameters selected by the user.

Replica broker uses some of core services such as *Replica Location Service(RLS)*, *Network Monitoring Service* (NMS) and *Data Transport Service* (DTS). These core

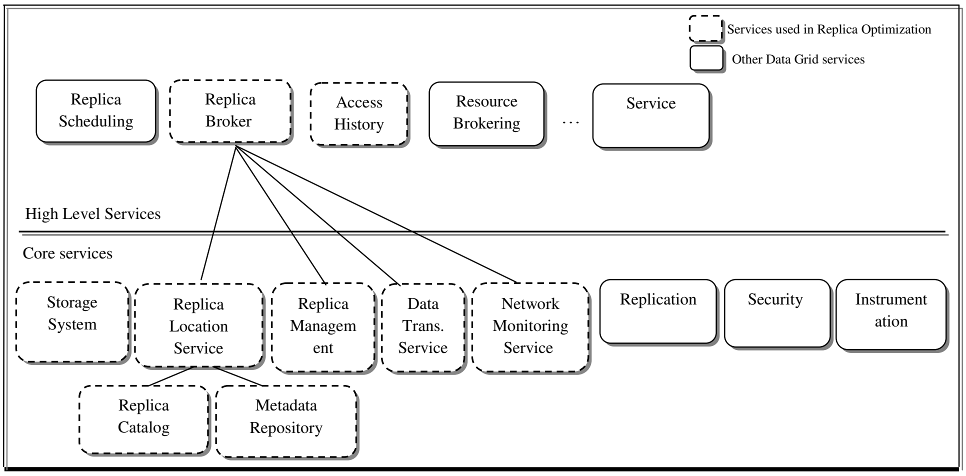

Figure 1.3: Data Grid architecture

services are used to collect information regarding replica providers (advertized QoS) and user preferences (QoS demands) to guide selection then transfer the required file from a set of replica provider alternatives.

The proposed broker (2SOB) brings in a novel approach of using the Data Mining techniques of association rules to discover a suitable set of replica providers, and Grey-based Rough Set Theory to filter the replicas that match user/application requirements. The main objective of the proposed broker 2SOB is to find multiple replica providers that are *stable* (uncongested network links) and have a *close match* to the user's requirements.

The term Data Mining refers to non-trivial extraction of implicit, previously unknown, and potentially useful information from data. This encompasses a number of technical approaches, such as clustering data summarization, classification, finding dependency networks, analyzing changes, and detecting anomalies [Puj01]. Many of these approaches have been used previously in replica selection techniques for grids in the literature. All these methods in general are not so convenient for broker design to achieve replica selection in Data Grids environments. This is due to large computational complexity and high I/O operation requirement. These methods generally differ with each other in their architecture, computational cost, and quality.

Presently in literature, the Rough Set approach and also fuzzy-based approaches have been used to deal with the suppliers selection problem under uncertainty [ZZLL09].

In the first phase, (*Coarse-grain phase*) of our proposed broker, we use 'finding dependency networks' approach of Data Mining as a replica discovery strategy. Coarse-grain strategy uses association rules concept to find groups of associated service providers (replica provider sites) which have lower rates of packet drops with good latency at the time of data file transfer [KR01].

In the second phase, called *Fine-grain phase*, we use *Grey-based Rough Set Theory* as a filtering strategy. In the Fine-grain phase, Grey values are used to deal with non-numeric values of user or provider attributes to solve the uncertainty issue. The Lower and Upper approximations and also Reduct concepts are applied on numeric and non-numeric variables to find set of associated providers that have the closest match to the requirements of the user/application. The set of associated replica providers with closest match to the user request are used to simultaneously send different parts of the required files.

1.1 Example of Data Grid application

The *Large Hadron Collider* (LHC) is a good example of scientific collaboration where massive number of computing elements is used to process, store and share extremely large data collections. This is achieved by building Internet-based distributed computing platforms such as, *LHC Computing Grid* (LCG). As it is shown in Figure 1.4, LCG is organized into four levels, or 'tiers'. Tier 0 is *CERN*. CERN is the European Organization for Nuclear Research [EM06, CER], central computer, which distributes data to the eleven Tier 1 sites around the world. The Tier 1 sites, in turn, coordinate and send data to Tier 2 sites, which are centers that provide storage capacity and computational analysis for specific tasks. Scientists access the stored data through Tier 3 sites, individual computers operated at research facilities with different types of service brokers.

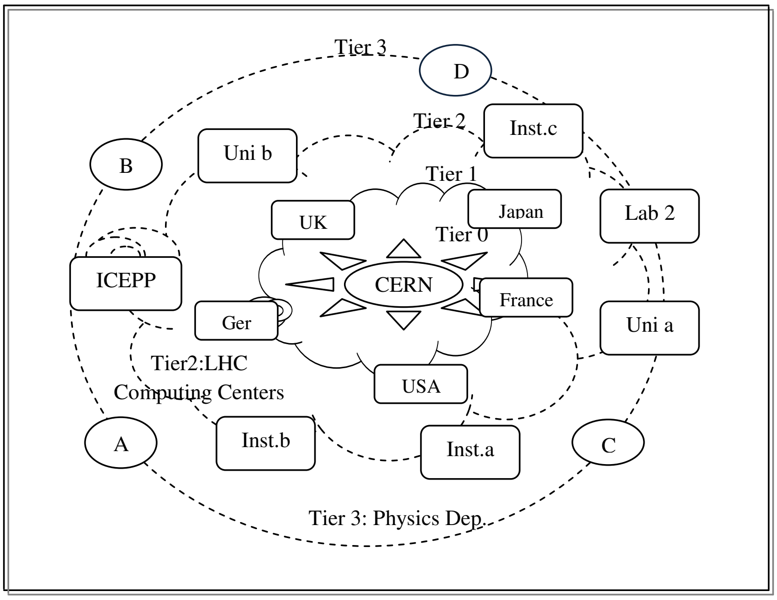

Figure 1.4: LHC Computing Model

EU Data Grid was designed to handle large volumes of data which would be put out by the LHC and other massive data producing programs [EM06]. In LHC, the data that has passed the triggering phases and been stored on tape is replicated into additional sites around the world for easier access and redundancy using the replication mechanisms of data grid as it is shown in Figure 1.5. CERN provides Data Grid applications. These applications allow people who work in the particle physics field to invent and run simulations as well as generate, test, and re-test experiments many times. This is possible through the construction of LCG. The physics experiments such as, ATLAS and CMS are expected to produce several petabytes of data per year. The consumers of experimental physics data and metadata will number in the hundreds or thousands. These users are distributed at many sites worldwide. Because of the geographic distribution of the participants in a particle physics experiment, it is desirable to make copies or replicas of the data being analyzed to minimize the access time to the data. Figure 1.5 shows the expected replication scheme for the physics data sets generated by the CERN Large Hardron Collider. Files are replicated in a hierarchical manner, with all files stored at a central location (CERN) and decreasing subsets of the data set stored at national and regional data centers [Hol00, HJMS+00].

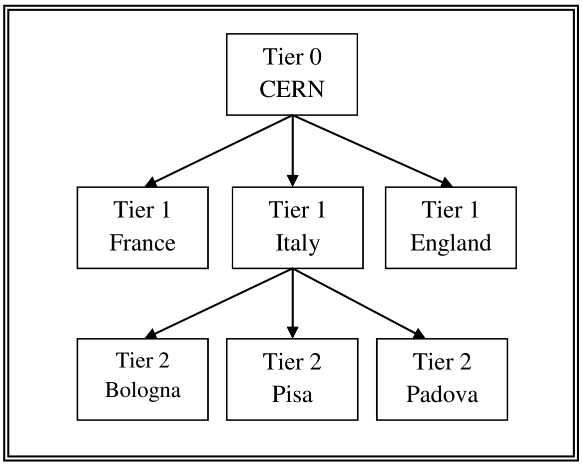

Figure 1.5: Scheme for hierarchical replication in CERN [LL02]

In the next sections, an overview of the fundamental data management services are presented.

1.1.1 An abstraction of a Data Grid

In a Data Grid there are two kinds of resources to be managed which are:

- **A Grid Data (GD):** It is any kind of data that can be located, transferred, replicated and manipulated. User/application clients' services should be able to access dispersed GD, independently from its physical location, through a *Data Grid Management System* (DGMS) [Jag06]. A DGMS is a software system used to manage Data Grids through the use of multiple abstraction mechanisms that hide the complexity of distributed data and heterogeneous resources. This naming capability allows users to refer to specific data resources in a physical storage system using a high level logical identifier.

- **A Grid storage space (GSS):** It is a storage space shared between multiple Virtual Organizations (VOs), and managed by a *Grid Storage Element* (SE). *Storage Resource Manager SRM* [DD09]) is an example of SRM. SRM is an interface to mass storage systems, providing a uniform control interface and enabling the Grid to efficiently use the storage. It's not necessary for a GD to be stored in GSS only, while a GSS may also contain data that can not be relocated, viz. are not GD.

1.2 Data Grid Layers

A data grid, like a classic computational grid, is designed to be mechanism and policy neutral, to have uniform access to grid resources and to be compatible with other types of grid infrastructures. The components of a Data Grid can be organized in a layered architecture as shown in Figure 1.6. This architecture follows from similar definitions given by Foster [VTF01a]. Each layer builds on the services offered by the lower layer in addition to interacting and co-operating with components and the same level (eg. Resource broker invoking VO tools). We can describe the layers from bottom to top as below:

- **Grid Fabric:** Consists of the distributed computational resources (clusters, supercomputers), storage resources (RAID arrays, tape archives) and instruments (telescope, accelerators) connected by high-bandwidth networks. Each of the resources runs system software such as operating systems, job submission and management systems and Relational Database Management Systems (RDBMS).
- **Communication:** Consists of protocols used to query resources in the Grid Fabric layer and to conduct data transfers between them. These protocols are built on core communication protocols such as TCP/IP and authentication protocols such as PKI (Public Key Infrastructure), passwords or SSL (Secure Sockets Layer). The cryptographic protocols allow verification of users' identities and ensure security and integrity of transferred data. These security mechanisms form part of the Grid Security Infrastructure (GSI) [FKTT98]. File transfer protocols such as GridFTP (Grid File Transfer Protocol), among others, provide services for efficient transfer of data between two resources on the Data Grid. Application-specific overlay structures provide efficient search and retrieval capabilities for distributed data by maintaining distributed indexes.
- *Data Grid Services:* Provides services for managing and processing data in a Data Grid. The core level services such as replication, data discovery and job submission provide transparent access to distributed data and computation. User-level services such as resource brokering (selection of resources for a user based on his requirements) and replica management provide mechanisms that allow

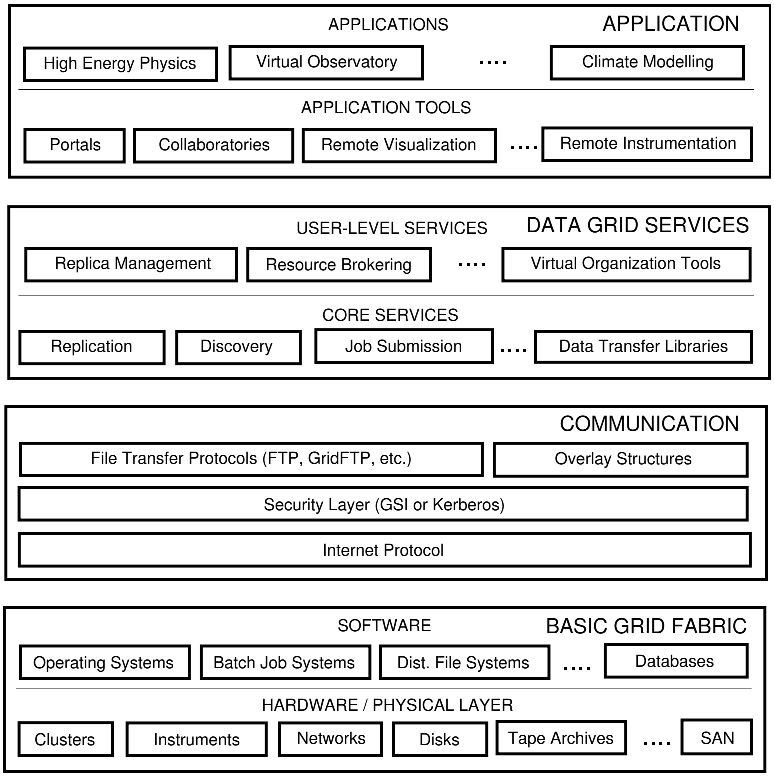

Figure 1.6: A Layered Architecture of Data Grid [VBR06]

for efficient resource management hidden behind initutive commands and APIs (Application Programming Interfaces). VO tools provide easy way to perform functions such as adding new resources to a VO, querying the existing resources and managing users' access rights.

- **Applications:** Specific services cater to users by invoking services provided by the layers below and customizing them to suit the target domains such as high energy physics, biology and climate modeling. Each domain provides a familiar interface and access to services such as visualization. Portals are web interfaces that provide single-point access to available VO services and domain-specific applications and tools. Collaboratories (Kouzes et al., 1996) have similar intent and also provide applications that allow users to conduct joint operations with their colleagues [VBR06].

There is a set of functionalities that every Data Grid should be able to perform which are:

- Ability to search through numerous available datasets for the required dataset

- Ability to discover suitable data resources for accessing the data and computational resources for performing analysis
- Ability to select suitable computational resources and process data of them
- Ability for resource owners to manage access permissions

1.3 Data Grid architecture

The requirements of the design led to a layered system architecture for the middleware that extents the classic grid middleware architecture as shown in Figure 1.3 similar to the one presented in [CFK+00]. The high level view shows that there are two layers: a lower layer for core services and higher layer services built on top. The lower level core services seek to abstract heterogeneous storage systems to have uniform ability to read, delete, create, and modify file instances. The core services also contain a metadata service that works to manage information about the data as well as replicated data. The key difference between a data grid and a classical grid was the proposal of the replica system and its subsystems (pointed components in Figure 1.3. These systems sit on top of the resource management system. The replica management controls the process of creating and deleting replicas which are essentially copies of data. The replication selection component is responsible for choosing specific data or file to be replicated (note this is different to choosing the data source at which a copy of a file or data maybe reside) [LAB06].

Data Grid is used by the scientific applications, such as High-energy Physics and Climate Modeling. These applications require two fundamental data management components, upon which higher-level components can be built:

- **Data Transport Mechanism:** *GridFTP*, is a universal data transfer protocol used for grid computing environments. It is a reliable, secure, high-performance data transfer protocol for use in wide area environments. Ideally, this protocol would be universally adopted to provide access to the widest variety of available storage systems.

- **Replica Management System:** For managing multiple copies of files and collections of files, including services for registering and locating all physical locations for files and collections as shown in Figure 1.3.

The Higher-level services that can be built upon these fundamental components include:

1. **Replication:** It is reliable creation of a copy of a large data collection at a new location

2. **Selection:** It is to select the best replica for a data transfer operation based on performance estimates provided by external information services; and automatic creation of new replicas in response to application demands

These two fundamental components of Data Management architecture, or Data Grid are explained in the next subsections.

1.3.1 Data Transport Mechanism

Applications that require to access data stored in different storage systems need a protocol for data movement. Data Grid has a transfer and access protocol called GridFTP that provides secure, efficient data movement in Grid environments. This protocol, which extends the standard FTP protocol, provides a super set of the features offered by the various Grid storage systems currently in use. Nowadays, Data Grids use GridFTP as a common data access protocol that would be mutually advantageous to grid storage providers and users. Storage providers would gain a broader user base, because their data would be available to any client, while storage users would gain access to a broader range of storage systems and data. The following are the main features of GridFTP [gloc]:

- **Parallel data transfer**: Multiple TCP streams to improve bandwidth over using a single TCP stream. Parallel data transfer is supported through FTP command extensions and data channel extensions

- **Secure data transfer**: Grid Security Infrastructure (GSI) and/or Kerberos authentication are used to support User-controlled settings of various levels of data integrity and confidentiality. This feature provides a robust and flexible authentication, integrity, and confidentiality mechanism for transferring files

- **Third-party control of data transfer**: Support for managing large data sets for large distributed communities. It provides third-party control of transfers between storage servers.

- **Striped data transfer**: GridFTP supports striped data transfers. Capabilities to partition data across multiple servers to improve aggregate Bandwidth

- **Partial file transfer**: GridFTP supports transfers of regions of a file, unlike standard FTP that requires the application to transfer the entire file

- **Reliable data transfer**: Fault recovery methods for handling transient network failures and server outages and for restarting failed transfers

- **Manual control of TCP buffer size**: Support for achieving maximum bandwidth with TCP/IP

- **Integrated instrumentation**: Support for returning restart and performance markers

1.3.2 Replica Management System RMS

This section explains the architecture of Replica Management System (RMS). It is a layered architecture. RMS is called a data replication selection mechanism, it allows users to create, register, and manage complete and partial copies of data sets, or to update new replica versions if the original datasets are modified. Replica management is an important issue for a number of scientific applications. For example, consider a data set that contains petabytes of experimental results for a particle physics application. While the complete data set may exist in one or possibly several physical locations, it is likely that many universities, research laboratories or individual researchers will have insufficient storage to hold a complete copy. Instead, they will store copies of the most relevant portions of the data set on local storage for faster access. In brief the services provided by a replica management system include:

- creating new copies of a complete or partial data set

- registering these new copies in a Replica Catalog (discussed in Section 1.3.2.1
- allowing users and applications to query the catalog to find all existing copies of a particular file or collection of files
- selecting the 'best' replica for access based on storage and network performance predictions provided by a Grid information service

In Data Grid architecture there are two types of services: Core-level and High-level services which are explained in the next sections.

1.3.2.1 Core-level services

The most important low level services which are used to build our proposed broker are:

- **Replica Catalog (RC)**: RC is the lowest level of RMS layers. It contains information about locations of datasets and associated replicas and the metadata associated with these datasets. RC is used in Data Grid, whenever new copies of a complete or partial data set are created. The purpose of the replica catalog is to provide mappings between logical names for files or collections and one or more copies of the objects on physical storage systems The catalog registers three types of entries: logical collections, locations and logical files. A logical collection is a user-defined group of files. We expect that users will find it convenient and intuitive to register and manipulate groups of files as a collection. Location entries in the replica catalog contain all the information required for mapping a logical collection to a particular physical instance of that collection. Each location entry represents a complete or partial copy of a logical collection on a storage system. One location entry corresponds to exactly one physical storage system location.

  The location entry explicitly lists all files from the logical collection that are stored on the specified physical storage system. Despite the benefits of registering and manipulating collections of files using logical collection and location objects, users and applications may also want to characterize individual files. For this

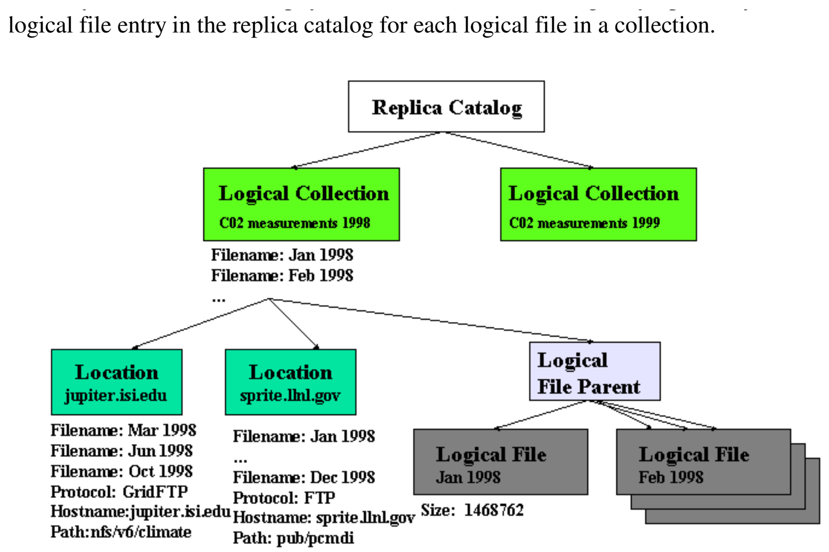

Figure 1.7: A Replica Catalog for a climate modeling application [gloc]

purpose, the replica catalog includes optional entries that describe individual logical files. Logical files are entities with globally unique names that may have one or more physical instances. The catalog may optionally contain one logical file entry in the replica catalog for each logical file in a collection as it is shown in Figure 1.7.

- **Metadata or data about data**: It is an information that describes the datasets. A metadata system could contain attributes such as name, time of creation, size on disk, and time of last modification. Metadata may also contain specific information such as details of the process that produced the data

- **Network Monitoring**: Performance monitoring is another essential service for data grids. Applications and resource brokers need to monitor the state of the grid, including network bandwidth, space available at storage servers, the load on computational resources, and the progress of data transfers. Based on this state information, the application or broker can make improved scheduling and resource selection decisions. It is to provide an information base from which to perform intelligent replica selection [gloc]

1.3.2.2 High-Level Services

As shown in Figure 1.3, in RMS the core Data Grid services can be used to construct a variety of higher-level services which are:

- **Replica Management**: Replica management is the process of creating or deleting replicas at a storage site. Most often, these replicas are exact copies of the original files, created only to harness certain performance benefits. A replica manager typically maintains a replica catalog containing replica site addresses and the file instances.

- **Replica Selection**: Replica selection is the process of choosing a replica from among those spread across the Grid, based on some characteristics specified by the application. One common selection criteria would be access speed [VTF01a].

An application that requires access to replicated data begins by querying an application specific metadata repository, specifying the characteristics of the desired data. The metadata repository maintains associations between representative characteristics and logical files, thus enabling the application to identify logical files based on application requirements. Once the logical file has been identified, the application uses the replica catalog to locate all replica locations containing physical file instances of this logical file, from which it can choose a suitable instance for retrieval. The entity that identifies the suitable instance of a replicated file based on application requirements is referred to as a broker. In effect, the responsibility of the broker is to map application requirements against storage resource capabilities. The following section is to discuss the storage broker.

1.4 Data Grid Resource Broker

As defined in Foster's "Physiology of the Grid" [ABC+07], a Grid service is any service used in the Grid environment that conforms to specific interface conventions accepted as standard throughout this Grid. However, the low-level fabric provided by *Web Services Resource Framework* (WSRF) is insufficient for developing complex-problem solving environments. Complex higher-level functionality is necessary to provide a seamless, transparent, manageable, reliable, efficient and secure common computing environment, in which a resource broker is an integral part [KS06]. As it is shown in Figure 1.6, the *Resource broker* (RB) is one of a user high level service in the middleware of Data Grid. RB is responsible for carrying out a set of tasks related to user job submission. These tasks include interacting with Replica Catalog (RC) to resolve logical data set names, finding preliminary set of sites for data transfer, job submission and management. The resource broker is a daemon process that listens to a TCP/IP socket for client requests. Upon client interaction, a thread is spawned to handle the client messages using a new available port. The RB master daemon acts as the preliminary broker for agent (an instance of RB) thread to handle individual client. In Chapter 3 the details of broker are explained.

1.4.1 Internal design of the Replica Broker

One of the most important operation of the previous sequenced operations is, selecting the best replicas using a Replica Broker. In this section the general description of our proposed Replica Broker is given. We followed the design of the GridBus Grid Service Broker [NVG+05] to design our broker.

In general the design of the broker is composed of three main sub-systems:

1. The user/application interface sub-system
2. The core-sub-system
3. The execution sub-system

The interface layer forwards the inputs of the broker. There are two main inputs:

A) A files description list, which corresponds with the job description created during composition of the Job, and represents the user request (job) that needs to be executed.

B) A replicas description list, which represents the available Grid replicas which have the requested files. The files description list (A) together with the delegated user's credentials is received from the client through the Web service interface of the broker. The replicas description list represents the current state of the available replica sites providers. In the broker core layer, the above inputs are converted into "jobs" and "replicas". The "job" is the abstraction for a unit of work have names of the required files needed to be moved to specific computational node. The "replicas" is the abstraction for the replica providers. Once the jobs are prepared and the replicas are discovered, the optimizer is started.

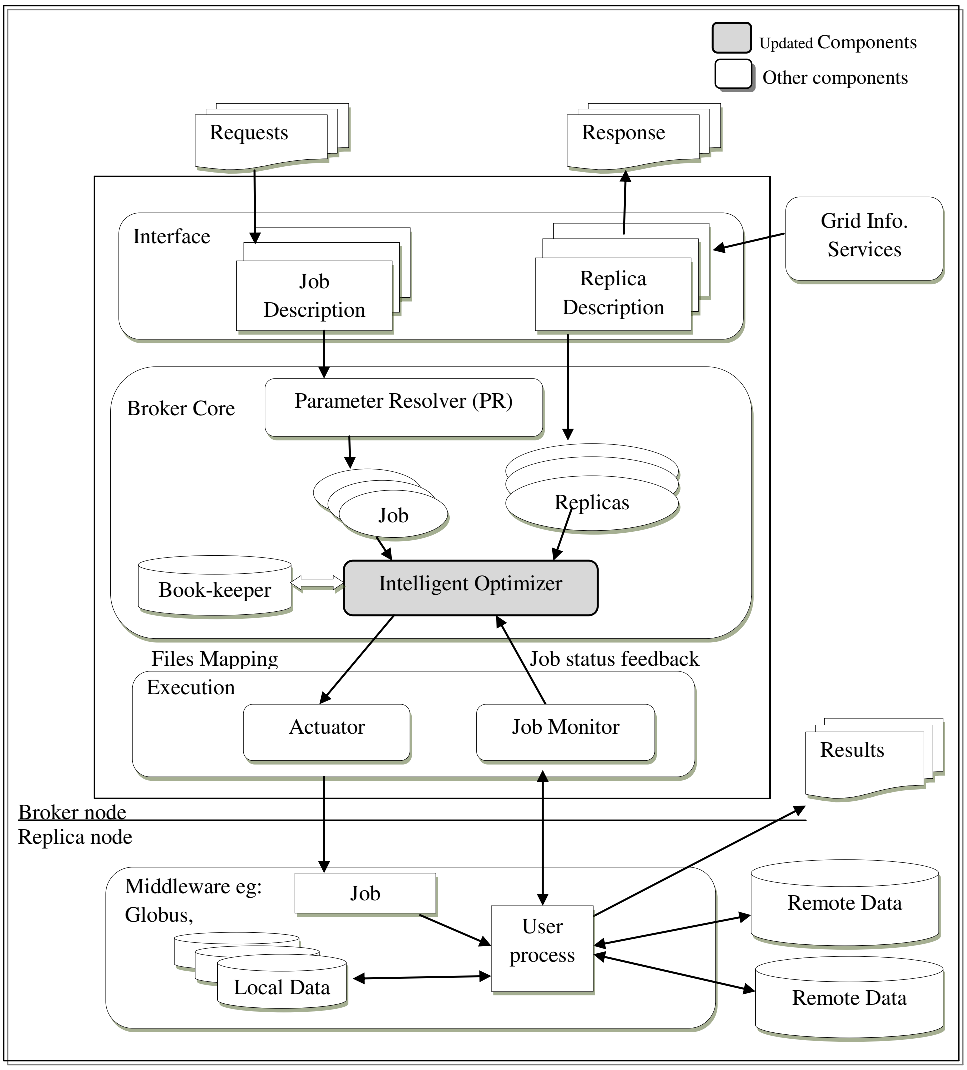

Figure 1.8: Enhanced GridBus resource broker architecture

The optimizer selects a best replica provider(s) based on its selection algorithm. The required files have to be transferred from the selected replica(s) (i.e. submits jobs of transferring the required files using the actuator component in the execution sub-system). The actuator is a middleware specific component, which dispatches the job to the remote storage Grid node. On receiving a job submission, each selected replica provider uses its own associated job-manager to actuate the middleware specific job-submitters). The job-monitor updates the book-keeper by periodically monitoring the jobs using the services of the execution sub-system. As the job gets completed, the job manager takes care of cleaning up and gathering the output of the jobs. Figure 1.8 illustrates the architecture of the internal design of of GridBus broker and shows the place of our replica broker which is called Tow phased Service oriented Broker (2SOB), works instead of the Intelligent Optimizer. The details of our proposed broker is given in Chapter 3.

1.5 Motivations and Objectives

The literature shows that the brokering strategies attached lot of research to come up with several strategies in the past. We observed that few of them applies some of soft computing techniques for only replica selection problem, but did not give proper attention to several other issue like scalability, availability, etc as given below. This motivated us to give up within the design of a broker which address these issues.

The main objective of the thesis is to design a scalable Replica Selection Broker, the scalability of our broker is defined by making the broker as a (listener). It listens to a specific port of TCP/IP. Listener initiates an agent (instance of broker) for each job. The agent runs on different site. Whenever the job finishes, all its reserved resources are released. The proposed broker should have the following characteristics:

1. **Scalability:** Replica selection process is scalable for large transaction volumes

2. **Reliability:** Choose the **set** of associated links of replica providers

3. **Stability:** Extract the most stable network links of the replica providers

4. **Uncertainty:** Allow for the vagueness and uncertainty between advertised and requested resources

5. **Availability:** Ensure that the required files are available in the selected providers

6. **Accuracy:** Select the closest match between users requests and replica providers services

7. **Efficiency:** Reduce the time latency of both, searching and file transfer during selection process

8. **Easy for deployment:** Built on the available core services, so it is easy to use without the need to change the infrastructure

Problem Statement: Design a scalable, reliable, stable, uncertain, available, accurate, efficient, and easy for deployment replica selection broker for Data Grid jobs using soft computing techniques.

1.6 Research Problems Related to the Objectives

A high level service oriented broker is needed for replica selection task in Data Grids. In this thesis we propose a scalable architecture of a service oriented broker. It is helpful in improving the efficiency of execution of jobs by reducing latencies to access data. In the design of the Two phased service oriented broker (2SOB), we had considered the following problems, all of which are with respect to current literature:

Problem 1: Scalability: In Data Grids a central manager for replica selection that performs matches against clients and replicas is inefficient (not scalable). How can we propose a design that is scalable?

Problem 2: Reliability: In the previous work on replica selection techniques, a single replica provider is chosen, when a selected provider fails, the job fails. How can an alternative choice be readily available?

Problem 3: Stability: In the previous work on replica selection techniques the network link of the selected replica provider may not be stable. This leads to an increase in the number of lost data packets during transportation. Can we choose sites with more stability?

Problem 4: Uncertainty: In most cases, attributes used to describe the services of replica providers or user requests are described using a single numerical value, were and never described by non-numerical linguistic values. In practice there is a large degree of uncertainty and imprecision in the attribute description, For instance, Security attribute can be expressed by one of these linguistic values, "Very Good", "Good", "Bad", etc.

Problem 5: Availability: In most of the previous strategies, the selected providers might or might not have the required file(s) when the transfer request was made. This situation occurs because some of previously suggested methods use outdated data for decision making.

Problem 6: Deployment: A broker is said to be a high level service layering on set of core services. Some selection strategies in the literature need extra information to do the selection. This information might be needed to add extra core services. In such a case the deployment of new broker needs to make changes in the Grid middleware. This is not easy to deploy.

Problem 7: Inaccurate choice (mismatched selection): Selection is not good of quality where the selected provider is not the best provider according to the user request i.e., deficiency in the quality of service using previous selection methods.

Problem 8: Inefficiency in selecting set of providers: Using the previous strategies, to select W providers, the strategy needs to re-run W times. For every run a provider is selected.

For these problems we attempted solutions which are explained through out the thesis and summarized in Section 1.7.

1.7 Contributions in Summary

The contributions made by this thesis, demonstrating the effectiveness of the *Service Oriented Broker for Replica Selection in Data Grid*, are delineated below:

Problem 1. A scalable replica selection broker is designed. In our architecture, for each job an agent of two phased service oriented broker (2SOB) is being initiated. 2SOB is a decentralized replica brokering strategy wherein every client that requires access to a replica performs the selection process rather than a central manager performing matches against clients and replicas. The details are explained in Chapter 3 of the thesis, and the work is published in [AWNR11].

Problem 2. 2SOB generates a list of alternative replica providers. In case when any of the selected provider fails it is replaced by one from the candidate set of alternate providers. It increases quality of service for a user with a sustainable set of data resources. This is done using an association rules concept of data mining approach. The strategy is explained in Chapter 4 in the Coarse-grain phase, and the work is published in [AWNR10d].

Problem 3. 2SOB generates a list of replica providers that favor stable providers (providers network links are stable) over unstable providers. It increases quality of service for a user with a stable data providers. This is done using an association rules concept of data mining approach. The strategy is explained in Chapter 4 in the Coarse-grain phase, and the work is published in [AWNR10b].

Problem 4. 2SOB deals with the uncertainty issue between advertised resources and requested resources using linguistics variables. A Grey-based Rough Set Theory is used as a selection strategy that works with uncertainty values. Dealing with Grey based rough set theory the resources matchmaking and ranking process in 2SOB is done. It is explained in Chapter 5 of the thesis, and the work is published in [AWNR10d].

Problem 5. A scalable strategy, known as Efficient Set Technique (EST) is used in 2SOB. EST selects set of replica providers that can share transfer of different sizes of files in constrained time. With this strategy there is no possibility of non-existence of the file(s). This is because the EST depends on the current list of providers with the help of the replica location service (RLS). Using EST the 2SOB can deal with different files sizes. An association rules strategy is used to extract the stabilized links that connect replica providers sites with computing site. The required amount of data (Terabyte or above) is divided to be transferred simultaneously from the stable set of providers as shown in Figure 3.7. This is explained in Chapter 2 and 4 of the thesis, and the work is published in [AWNR10a], [AWN+10a].

Problem 6. 2SOB is easy to deploy and used in Data Grids infrastructure. It is constructed by layering on a set of an existing core services, such as data transport, security, and replica cataloging. It enables organizations to improve productivity and reduce transfer costs through easy-to-deploy solutions. This is because 2SOB could work within the existing Data Grid infrastructure for increasing the delivering operational efficiencies. This is explained in Chapter 3 of the thesis, and the work is published in [AWNR11].

Problem 7. 2SOB uses lower approximation and Reduct concepts of Rough Set Theory. This concepts classifies replica providers according to the QoS attributes in the user request. This method is explained in Chapter 5 and 6 of the thesis, the work is published in [AWNR10c].

Problem 8. 2SOB selects W providers, so, there is no need to re-run the strategy because the output of selection strategy is a set of associated providers (BRS). W providers can be selected from BRS. This is explained in Chapter 4 of the thesis, and the work is published in [AWN+10a].

1.8 Organization of the Thesis

The thesis has been divided into seven chapters, that are organized as follows.

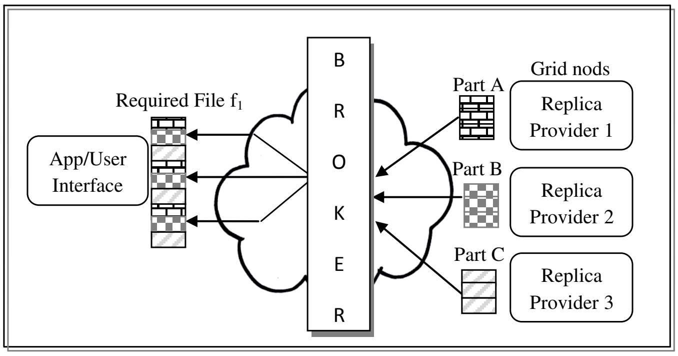

Figure 1.9: Concurrent bulk data transfer of file (f_1) from multiple providers

1. **Chapter 1:** Introduces Data Grid notions and some basic concepts about Brokers in Data Grids. These notions form a base that is used throughout the thesis

2. **Chapter 2:** Focuses on Taxonomy of replica selection techniques. Various approaches from literature are described

3. **Chapter 3:** Explains the architecture design for a Two phased Service Oriented Broker along with a description of its active components

4. **Chapter 4:** Explains the general concepts found in the Data Mining approach focusing on the association rules model and then explains in details the Coarse-grain phase. It ends with experimental datasets that were taken from the National Accelerator Laboratory, SALC where all Data Grid sites of the world (connected to the CERN project) are seen from CH.CERN.N20 [cot, pin]

5. **Chapter 5:** Focuses on explaining the general concepts of Grey value and Rough Set Theory, focusing on the approximations and reduct concepts and then explains in details the Fine-grain phase. It ends with providing results of applying the strategy on various synthetic data sets with two study cases

6. **Chapter 6:** Focuses on explaining the Rough set based Quality of Service Design for service provisioning in Clouds and provides results of applying various techniques on the case study

7. **Chapter 7:** Concludes the thesis with a brief review on contributions made and discusses scope for further research in the area.

1.9 Chapter Summary

In this chapter we have started with a background of Grids with some essential concepts related to data grids. Then the construction of Grid systems. Then we have briefly presented the taxonomy of grids. This is followed by example on Data Grid application. We take up then Data Grid Abstraction and Layers. The architecture and internal design of Data Grids is shown along with a description of its active components. A general introduction for a resource broker is then proposed. This is followed by research problems that thesis focuses and contributions. At the end we presented our publications and an Organization of the thesis.

Chapter II

Literature Survey

In this chapter we look at some of the approaches where grid resources are acquired and their selection strategies are studied. The aim of the study is to determine the common concepts and observe their performance.

2.1 Introduction

As we mentioned in Chapter 1, that Resource Broker is a decision making tool that determines how and when to acquire grid services and resources for higher level components. Several brokers are proposed and developed by different commercial companies and academic research groups. Some Resource Broker implementations includes Grid Lab Resource Management System (GEMS) [gria], Nimrod/G [Nim], Grid-Bus [LAB06], etc are already in use in diverse scientific areas and projects. Each one of the above tools has different advantages and integration constraints. However, at the time when this work was started, to the best of our knowledge none of the above brokers had a scalable optimizer which selects a set of replica providers that have uncongested network links and matches to closely user's requirements. The following sections give an overview of Data Grid job and file access patterns, these concepts are during literature survey.

2.2 Data Grid Job

In Data Grid a user/application typically submits a job to the Resource Broker by job submission interface. Data Grid job contains logical names of files which user/application wants to analyze in a specific time constraint. The required files might be available locally or possibly at different Grid sites providers. These files need to move from providers sites to another site (computing site) where computational resources are used to perform operations on the data located within the files.

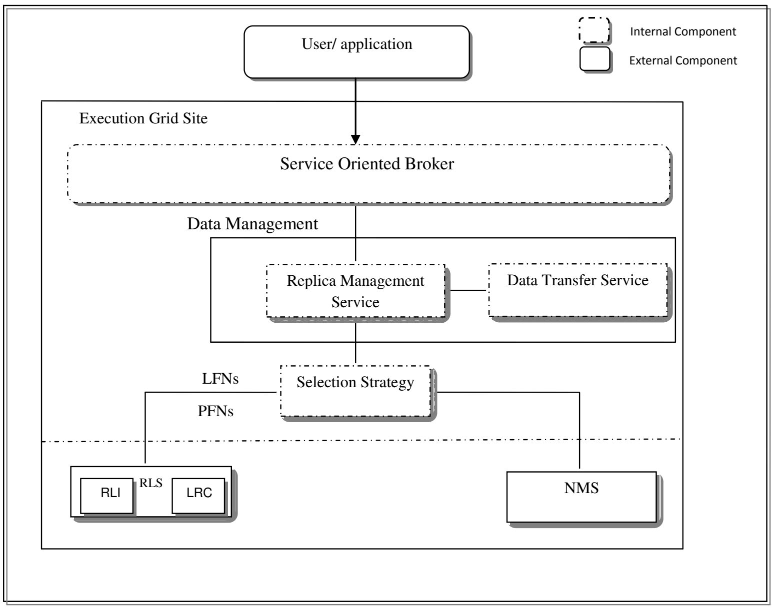

Figure 2.1: Data Grid components [LAB06]

To decide which replica provider site is chosen, Resource Broker uses selection mechanisms to take a decision for each job based on the current state of the Grid resources (location of data, network load, cost of file and so on).

To access files, two types of references may be used: a *Logical File Name* (LFN) and a *Physical File Name* (PFN). A LFN is an abstract reference to a file, which is independent of both where the file is stored and how many replicas exist. A PFN refers to a specific replica of some LFN, located at some particular site. LFN will have one PFN for each replica on the Grid. A job usually requests the LFNs for data access. By looking up the catalog, the list of PFNs will be found using a replica location service, and optimization then proceeds by finding which PF can be accessed in the shortest time. Thus, the Replica Selection Strategy (RS) is designed to optimize the selection decision of one or more replicas on the Grid sites, thus helping to maximize the job throughput as it is shown in Figure 2.1.

In GridBus broker for example, the *Replica Optimization Service* (ROS) is used as a selection service. ROS is an internal service component of the Replica Management Services which depends on the three underlying services:

- **The Replica Location Service (RLS):** RLS is used to determine the physical addresses of the required files using Replica Location Index (RLI) and Local Replica Catalog (LRC).

- **The Network Monitoring Services (NMS):** NMS is used to determine the estimated cost for file transfer over the WAN.

- **The Storage Element (SE):** SE is used to determine the access cost for a specified file, i.e. how much time it takes to make the data accessible. The SE may have various mass storage media underneath where the data resides and depending on what kind of media the file is on (tape, CDROM, disk) it may take some time to prepare (stage) the file for access.

Based on these access time estimators ROS chooses the 'best' replica among many as a function of the accessing site [LAB06]. The selection is done in two ways:

1. Firstly the ROS provides an estimate of the access cost for a job to run at a particular site, i.e. the time it would take to access all the files required for the job. This information is used by a Resource Broker to schedule the job to the optimal Grid site as shown Figure 2.2

2. Secondly, the ROS is responsible for optimizing file access as jobs are running and this is carried out in two stages: *Short-term optimization* and *Long-term optimization* which are explained in the next section

2.3 Files Access Patterns

As mentioned above, a job will typically request a set of logical file name(s) for data access. The order in which the files are requested is known as access pattern. The common access patterns are:

1. **Sequential Access**: In this pattern the files are accessed in the order that has been stated in the job configuration file

2. **Random Access**: In this pattern the files are accessed using flat random distribution

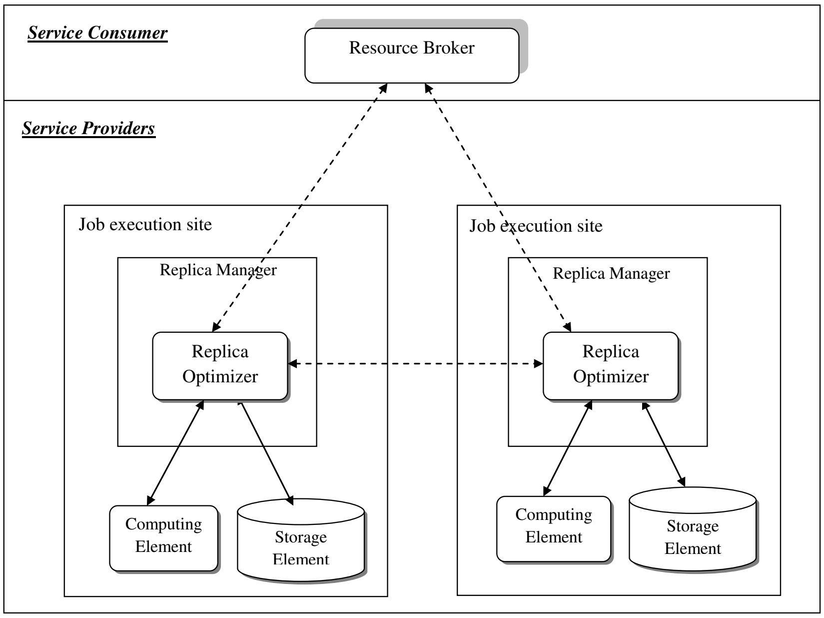

Figure 2.2: Data Grid component interactions in GridBus [LAB06]

3. **Unitary Random Access**: In this pattern the file requests are one element away from previous file requests but the direction will be random

4. **Gaussian Random walk Access**: In this pattern the files are accessed using a Gaussian distribution These file access patterns are shown in Figure 2.3

The next section is to give an overview of some of previous replica selection methods.

2.4 About replica selection strategies

This section presents some selection strategies which are proposed in the literature to improve the performance of a resource (replica) broker.

In the last few years there were two directions followed by researchers to reduce the execution time for data grid jobs. The first approach focused on reducing the time of selecting the best replica site. In this approach researchers proposed different selection strategies used in Replica Optimization Service Broker to enhance the selection process [LAB06]. The second approach was focused on dividing files into multiple parts and then co-allocation architecture was used to enable parallel downloading of data sets from multiple replica provider servers. The objective of

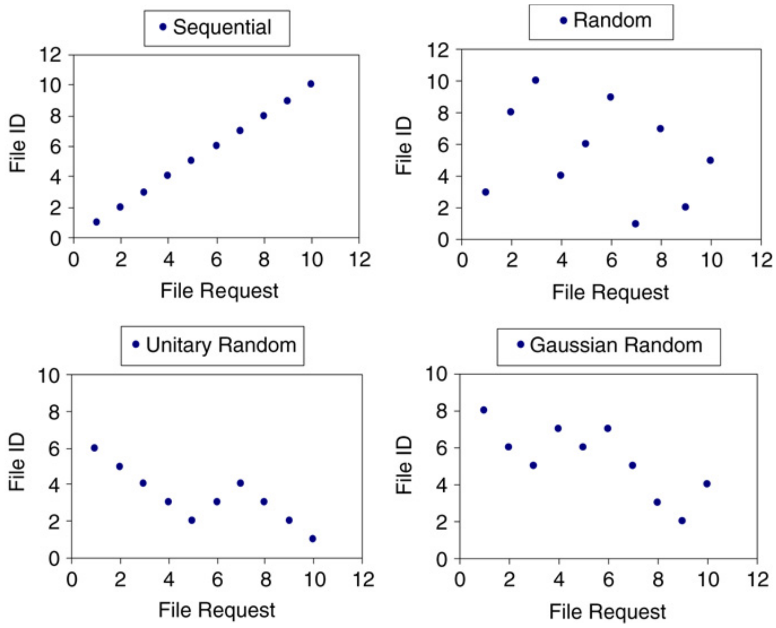

Figure 2.3: Various file access patterns [RAB08]

the second approach is to exploit rate differences among various computing site and replica provider links and also to address dynamic rate fluctuations by dividing files into multiple blocks of equal sizes to improve the performance of data transfer in Data Grids [YYCW06]. In our work a novel replica selection broker is proposed to reduce the total execution time of data grid job by selecting a set of uncongested network links of replica providers and the providers in the selected set have a close match to the user/application requirements.

J. Gwertzman and M. Seltzer [GS95b] and J. Guyton and M. Schwartz [GS95a] proposed replica selection approaches based on binding a client to the nearest replica, with respect to some static metric such as the geographical distance in miles and the topological distance in number of hops [cis]. However, as several experimental results [SBSV98, ZAZB00] show, the static metrics are not good predictors for the expected response time of client requests. The main drawback of both geographical and topological network metrics is that they ignore the network path's dynamic conditions.

From a different focus, R. Kavitha, and I. Foster [KF01], used traditional replica catalog based model; for each new request Replica Location Service is queried to get the addresses of replica's sites and then probe the network link using HopCount method to select the best replica. The drawback of this approach is that it depends on the number of hops that may not reflect the actual network condition such as Network Bandwidth and link's latency. On the other hand, Sudharshan et al. [FVS08, VS03, VTF02] contributed in many research results. In their work they used the history of previous file transfer information to predict the best site holding a copy of the requested file. When a file transfer has been made between two sites, the file size, the available network bandwidth, and transfer time are saved. Thus it can be used later for training and testing the regression model to predict the actual transfer time. In their work they showed that data from various sources can help in better predictions than data from one source. They achieved a better accuracy in file transfer throughput prediction by using data from all of these three sources: network data streams, file size, and past grid transfer information.

Several heuristic data replication schemes have been proposed in the literature [Hol00, BCCS+03, CZSS02, RF01, LA00, Aba04]. A dynamic replication scheme was proposed based on a variety Vickery auction for an economy based replication solution which is discussed in [BCCS+03].

F. Corina and M. Mesaac [FM03] and Ceryen and M. Kevin [BCCS+03] used different algorithms such as greedy, random, partitioned and weight algorithms in the selection engine. Rashedur et al. [RBA04] exploited a replica selection technique with the K-Nearest Neighbor (KNN) rule used to select the best replica from the information gathered locally. The KNN rule selects the best replica for a file by considering previous file transfer logs indicating the history of the file and those similar. This technique has a drawback as they mentioned in their paper: the misclassification will increase in case of the large file transfer and will cost more than a couple of small file transfer misclassifications. Especially in the Gaussian random access pattern the accuracy is the lowest. Another drawback in KNN is that one needs to save all previous instances (file requests) to use them to select the best replica site, which means it will take some time to search in the large history of data base and the result might or might not be correct.

H. Lin, et al [LAB06] have explored the effectiveness of economy-based resource management in data grids and proposed a policy for a data replication broker that improved replication. An economic data resource management system was shown to perform with marked improvements when using a variable pricing scheme that is based on the expected actions of a rational agent compared to a fixed pricing scheme. The policy displayed an effective means of improving the performance of the grid network traffic and was indicated by the improvement of speed and cost of transfers by brokers.

Rashedur et al. [RBA04] proposed a Neural Network (NN) predictive technique to estimate the transfer time between sites. The predicted transfer time can be used as an estimate to select the best replica site among different sites. Simulation results demonstrate that Neural Network predictive technique works more accurately than the multi-regression model, which was used before NN [FVS08, VTF02, VS03]. Nevertheless, NN technique does not always give the right decision because the copy of the file may no longer be available in the predicted site (this is a common occurrence in grid), so in this case the Traditional Model has to be used.

A. Jaradat et al. [JSA09] proposed a new approach that utilizes availability, security and time as selection criteria between different replicas, by adopting K-means clustering algorithm concepts to create a balanced (best) solution. The best site does not mean the site with shortest time of file transfer, but the site which has three accepted values: security level, availability and time of file transfer. The problem of using K-means is: the selection is being trapped in local clustering centroids.

The problem of finding an optimal replica allocation at a given state of the grid, i.e. an allocation that has minimum transfer cost for a given read-write pattern, has been shown to be NP-complete by Ouri Wolfson and Amir Milo [WJ95].

Vazkhudai et al. [VS02, VS03] show that disk I/O effects on the transfer time in Grid environment so, all these criterion can be used during the selection process.

All previous strategies were looking for finding the best single replica site using different approaches. By not assessing the quality of service one receives from a provider, the consumer cannot be considered completely economically rational. From an economic perspective this blind side to quality of service allows for lower quality services to be offered at lower costs – slowly forcing a global reduction in grid service quality over a time as higher quality services become ignored and unprofitable. Factors of transfer quality such as node-to-node link reliability, actual transfer times, and server reliability have been ignored so that future broker decisions will make the same choices.

So summarize, following are the drawbacks of the previous methods:

1. History file does not reflect the recent information; it is outdated (K-Nearest Neighbor rule (KNN))

2. Bandwidth alone and Hop counts alone does not describe the real network condition (Traditional Method (TM), Neural Network (NN) and KNN)

3. In the classification method, the misclassification will increase in case of transferring large files and using a Gaussian Random file access pattern

None of the previous methods reflect the real picture of network links. In the next two sections, some of alternative Replica Selection Strategies and Brokers are explained in details for clarity of comparison with our proposed Broker.

2.5 Alternative Replica Selection Strategies

As we noted, when different sites hold replicas of a particular file, the access latency is minimized by selecting the best replica. In the following sub sections we will discuss few important replica section methods available in the literature with their drawbacks.

2.5.1 Replicas selection strategy in the Traditional Model

In Traditional Model (TM) whenever the job is submitted to the Broker, broker checks whether the files are locally available in the local storage element or not. If the files are locally available, the computing element immediately accesses them, otherwise, it gets the files from different distributed sites [gloc]. The conventional (traditional) method represents the short term replica optimization service and uses following steps:

1. Receives Logical Files Names (LFN)
2. Contacts the Replica Location Service (RLS) to get Physical Files Names (PFN)
3. Contacts Network Monitoring Service (NMS) to probe the links between the computing site and replica providers to get information about network links for all replica providers. Iperf [ipe] is an example of testing route tools in NMS in Data Grid
4. Ranks replica providers
5. Selects provider with highest Bandwidth (BW)

The replica site with the highest network bandwidth or with the least HopCount measurement will be selected as a best replica provider. *Replica Lookup Time* is the time spent in the above steps [ATS93]. In other words, the lookup time is the consuming time by consumed the replica catalog with the logical file names to get the physical file names and then probe the links between the requester site and replica sites using a testing route tools.

For several years many researchers proposed various strategies to improve the traditional selection process. In the next subsection discusses some of these methods. In our work, to improve the selection process further, which is discussed in Chapter 3, we propose a new high level brokering service with two phases of selection policies that is called *Two phased Service Oriented Broker* (2SOB).

2.5.2 Replica Selection Strategy using the Neural Network (NN)

In NN model authors reduced total execution time of Data Grid job by avoiding look up time. In this case replica provider sites are predicted by the trained neural network instead of probing the routes as Traditional Method (TM) does [ATS93]. The *file history* that contains information about previous transfers like file size, the available network bandwidth, and transfer time for at least 20 jobs is used as a trace data to train the prediction model. That means the trained NN selection model starts predicting best replica provider from the 21^{st} job index [RAB08]. Back-propagation algorithm is used to train the model. Datasets in the history file are presented to the neurons of the input layer so that, the output neuron can predict (estimate) the total transfer time needed to transfer file(s) from the replica providers sites to the computing site.

Their NN selection idea is: the trained model is used to predict total files transfer time with respect to three factors, previous grid transfer time, current network bandwidth, and the file size. The site with the lowest predicted transfer time is chosen as the best replica provider. The experiments show that, NN model works properly only when the bandwidth of the network links is the unique criteria for the network. However, this is not a realistic condition in the Inter-Data Grid environment. In general the latency metric is more effective than bandwidth metric to be used in the selection strategy. The drawback of NN model are:

1. It selects only a single replica provider (Scalability)
2. Selected provider may fails (Reliability)
3. Does not probe network links (Instability)
4. Works with a numeric attributes only (Uncertainty)
5. Contacts Replica Location Service to get the physical location of replica providers (Availability)
6. Predicting model starts after at least 20 Data Grid jobs (Not easy for deployment)
7. The selected site might or might not be the best site matching with the user requirements (Inaccurate choice)

8. It may or may not give a correct prediction. In case of wrong prediction the total time using NN model will not be the minimum (Inefficiency)

These drawbacks will be used during the comparison between our proposed broker and NN will be explained in Chapter 4.

2.5.3 Replica selection strategy using K-Nearest Neighbor rule model (KNN)

In this model the best replica provider is chosen without probing to the network links or contact replica location service, that means the lookup time is zero [RAB08]. In KNN model, when a new job is submitted, information of K previous jobs that are similar to the new one are considered to predict the best replica provider. In this model the predicted site may or may not have the file during the time when the request was made. The classification of the first case is called right classification; otherwise it is called a wrong classification.

The idea of KNN model is: use traditional method for the first thirty jobs selection. Information about these 30 jobs are saved and will be used to predict best replica provider for the next job (31^{st}). After each job, information such as: the file index, the destination site index number, the requesting site index number and the time stamp of the request is tabulated into history file to be used later to select best replica provider for job number 31^{st} [RAB08]. That means, at request number 31^{st} the selection strategy turns from the traditional method into KNN strategy. In KNN, when a request for a file arrives, all previous data of the history file is considered to find a subset with K of previous file requests (K tuples) similar to the new request. Then the K similar requests are used to predict the best replica provider to the request number 31st. The selected site using KNN model may not have the requested file. In this case the access latency will increase because the model will go back to use the traditional model, means it doesn't work properly with dynamic replica strategy In [RAB08], to evaluate the KNN strategy, they compare the result of the selection between KNN and TM strategies.

The drawbacks of KNN model are listed in the following:

1. It selects only a single replica provider (Scalability)

2. Selected provider may fail (Reliability)

3. Does not probe network links (Instability)

4. Works with a numeric attributes only (Uncertainty)

5. Does not contact Replica Location Service to get the physical location of replica providers (Availability)

6. Predicting model starts only after at least 30 Data Grid jobs (Not easy for deployment)

7. The selected site might or might not be the best site matching with the user requirements (Inaccurate choice)

8. It may or may not give a correct prediction. In case of wrong prediction the total time using KNN model will not be the minimum (Inefficiency)

This model works properly in two cases:

1. In the static replica strategy, a replica is presented until it's deleted by administrator or its time duration expired. In other words, the file is always remains available in replica site.

2. In the static network when there is no change in the network conditions.

In Inter- Grid architecture, even though the prediction is correct the selected site might or might not be the best site at the time of file transfer with respect to the network conditions.

The characteristics of all studied models are summarized in Table 2.1 for case of comparison.

No.	*Properties*	*TM* [gloc]	*NN* [RBA04]	*KNN* [RAB08]	*EST* [AWNR10a]
1	Scalability	N	N	N	Y
2	Reliability	N	N	Y/N	Y
3	Stability	N	N	N	Y
4	Uncertainty	N	N	N	Y
5	Availability	Y	Y	Y/N	Y
6	Easy for deployment	Y	N	N	Y
7	Accuracy	Y/N	Y/N	Y/N	Y
8	Efficiency	N	N	N	Y

Table 2.1: The limitations of various models

Followings are the limitations of previous replica selection models:

1. The replica site which is predicted by TM, NN and KNN may not achieve the least time to transfer the requested file
2. The predicted site may not have the copy of the requested file any more
3. None of the previously proposed methods took into account the Quality of Service (QoS) of user requirement like the maximum transmission unit of data packet and number of data streams that could be opened between two sites. (Our new technique takes all these points as criteria to select the best set of replicas)
4. The size of requested file(s) affects the efficiency of selection techniques. In NN and KNN, the efficiency decreases when large files are requested, but there is no impact on the size of the file in case of TM and EST
5. All of previous methods are unstable, which means they do not have alternative site when the selected site becomes down or that network link becomes congested during the transferring process. The previous methods need to run again in the failed cases
6. Current data grid selections are not scalable selection techniques and they select a single replica provider
7. All selection methods work with numeric variables not linguistic variables

2.6 Alternative Replica Broker in Data Grid

The well known resource broker in Data Grid are briefly explained in the next sections.

2.6.1 The GridBus replica broker

The GridBus resource broker is capable of submitting jobs to Globus' execution subsystem as well as to many other computational resources (e.g. Alchemi, Unicore, XGrid, and others). To achieve this purpose the broker translates the job description it receives into the format of the specific middleware controlling the selected resource. The translated job description also contains information for the Grid middleware regarding transfer of the required data and application. The transfer operation itself is carried out without any direct user interaction The GridBus broker's architecture is clearly structured and well designed from a software engineering point of view. Unlike many other resource brokers, this one does not require any particular information or security system. It is designed as a stand-alone piece of software, which can be integrated with various existent components. However, despite being the most reasonable choice, a careful investigation of the resource broker also revealed several drawbacks in its design, which prevented us from using the broker in its original state.

1. The broker was not service based, and needed to be installed on every client machine

2. The broker did not provide mechanisms for automated compilation of a list of the computational resources currently present in the Grid, but required the user to do so. This list should be compiled from the Monitoring and Discovery System4 [gloc] (MDS4), which is the implementation of an information system in the Globus Toolkit

3. The broker was unable to query MDS4 for obtaining the status of resources contained in the list [KS06]

4. The Replica Selection Decision of Gridbus as it is proposed by H. Lin et al. [LAB06], shows a single replica site that is selected based on a variable pricing scheme

5. Selection does not reflect the stability of the network links of providers

2.6.2 Nimord-G resource broker

Nimord-G [BAG00] is a resource broker that performs scheduling of parameter sweep task-farming applications on geographically distributed models. It supports deadline and budget based scheduling driven by market-based economic models. GridSim [BCC+03] simulates a Nirmod-G like Grid resource broker. It evaluates the performance of deadline and budget constrained cost and minimization scheduling algorithms. Though GridSim supports scheduling of jobs, it does not model anything related to data replication such as automated replica creation and/or selection.

2.7 *Taxonomy of Optimization Approaches for Resource Brokers in Data Grids*

Here, we explain the taxonomy of replica selection techniques which were used in the management system of replica selection process in Data Grids. The *Replica Optimization Service* (ROS) is used as a selection service. ROS is an internal service component of the Replica Management Services [LAB06]. ROS is classified into three types which are:

- **Scheduling Optimization:** Using long-term scheduling strategy, ROS is used to allocate the job site by considering suitability of the location of replicas and the computational capabilities of the sites.
- **Short-term Optimization:** When a job requests a file, the ROS is used to find the best replica on the Grid in terms of the cost of transferring the file to the local site where the job is running. It improves the decision of choosing the location from which replicas are fetched by considering available network bandwidth between sites.
- **Long-term Optimization:** Using long-term data access patterns, the ROS can create and delete replicas anywhere on the Grid according to its predictions of file usage across the Grid. Here, ROS decides which file should be retained as a replica [buy].

Our proposed broker is related to the second type of optimization, a short-term optimization as it is shown in Figure 2.4.

In the last few years the selection problem has been investigated by many researchers. The research work is divided into into two aspects: first was related to enhancement of the replica selection strategy. Second aspect is related to enhancing the file transfer strategy, by dividing the file into multiple parts. The first has been investigated by several research projects in the last few years. Different selection strategies in the literature were proposed in the broker to enhance the selection process [LAB06]. All previous strategies were looking for finding the best *single* replica site using different approaches.

Another side of the problem has been investigated by developing co-allocation architecture in order to enable parallel downloading of datasets from multiple servers. The objective of the second aspect is to exploit rate differences among various client-server links and to address dynamic rate fluctuations by dividing files into multiple blocks of equal sizes to improve the performance of data transfer in Data Grids [YYCW06].

According to the number of replica providers which are chosen by the selection strategy, Replica Selection services are classified into:

- **A Single replica provider:** To determine the best replica provider site among all available replica sites. This is just a single site with no other fall back options or ranking of other sites
- **Multiple replica providers:** To determine multiple replica sites among all available replica provider sites. The **Link Analyzes** by **Association Rules** is used to find the hidden relationships of network links of replica providers [AWNR10a]

It is known that the total execution time and the total cost (price) of obtaining a replica (file) are the most important factors required by a user/application. To optimize job in such a way that satisfy the requirement of the Data Grid's user/application, replicas selection strategies provide the best single replica site or the best set of replicas (M number of replica sites). This objective can be achieved through:

I **Minimizing (reducing) Cost:** The selection strategy selects the cheapest replica to be the best replica site using the answer from auction protocol [KG05]

II **Minimizing (reducing) Time:** The selection strategy attempts to reduce the total job execution time in different ways such as reducing look up time, reducing searching time or reducing transmission time.

III **Minimizing both:** The selection strategy reduces cost and time such that the

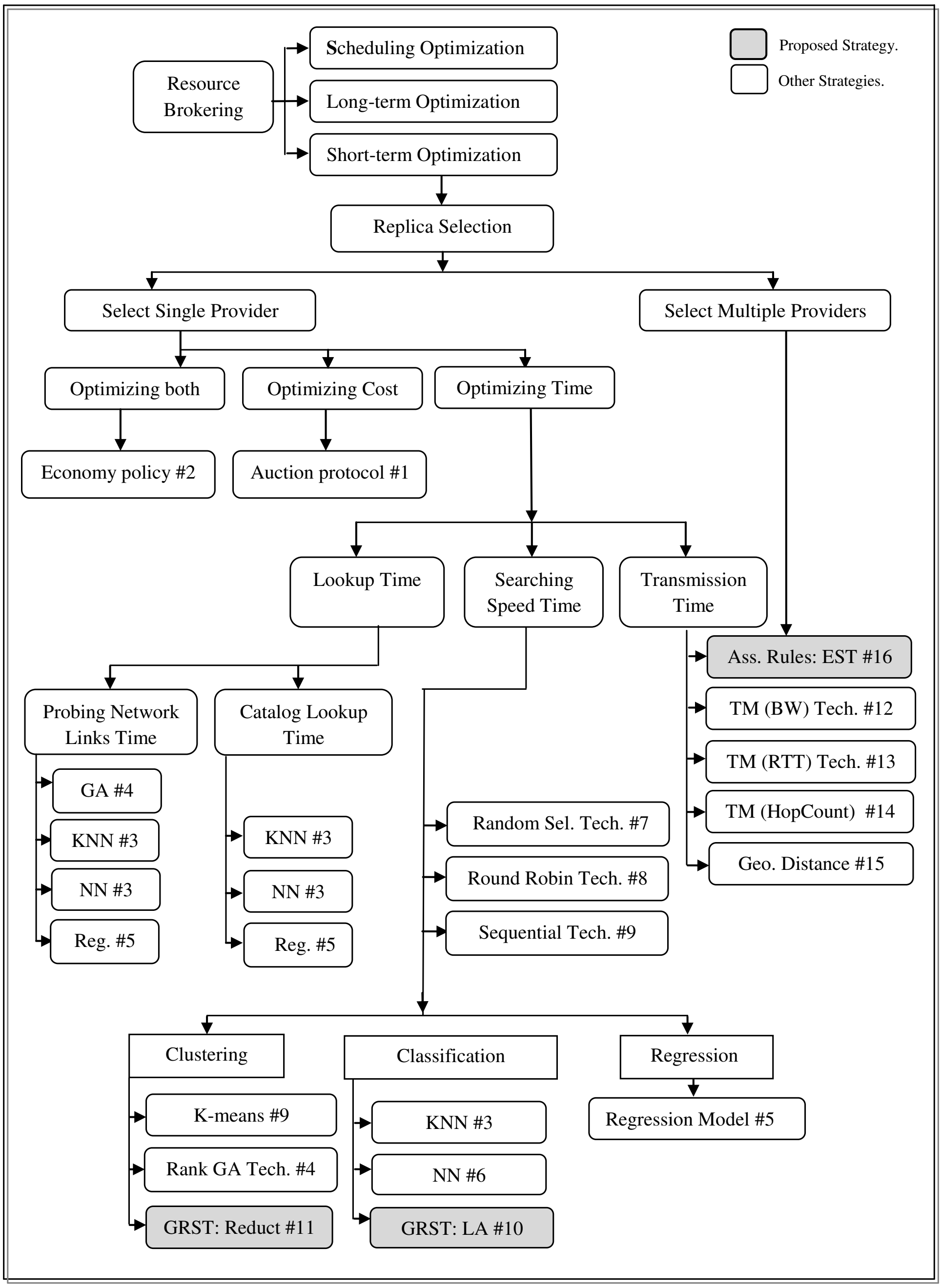

Figure 2.4: A Replica Selection Taxonomy (#1 [KG05], #2 [LAB06], #3 [RAB08], #4 [AAHJ10], #5 [VS03], #6 [RBA04], #7 [TM05], #8 [LVZ07], #9 [JSA09], #10 [AWNR10d], #11 [AWNR10c], #12 [VS03], #13 [VTF01b], #14 [Dir], #15 [SBSV98], #16 [AWNR10a])

answer from selection strategy is used in the economy-based data replication broker [LAB06]

To reduce the total execution time, the selection strategies of previous works are classified into three types according to the time period they wanted to minimize. The previous strategies had aimed to:

1. **Reducing the lookup time:** Usually most of selection approaches, minimize total execution time by minimizing time of replicas selection process. This can be done by reducing catalog lookup time and probing network links time. It is done either by using a classification technique such as K-Nearest Neighbor rules [RBA04] or predictive technique like Neural Network techniques [RAB08] or Genetic algorithm technique [AAHJ10]. All these methods attempt to avoid lookup time during predicting the total transferring time. All tried to enhance the traditional models which used the catalog look up time

2. **Reducing data transmission time:** To reduce the total time of file transfer, various selection methods were used such as, selection of the highest bandwidth among replicas or the shortest round trip time link, or the least number of hops (routers), or the shortest geographical distance. Recently association rules mining is also proposed by us [AWN+10a]

3. **Reducing the searching time:** In this process to get list of providers, the catalog must be looked up. That means, the catalog lookup time cannot be ignored. The searching time is the time that is consumed to find the replica provider who has the closest match to the user/application requirements. This time can be minimized by:

   (a) **Random selection:** The best replica is chosen randomly [TM05]

   (b) **Round Robin:** Round robin method is used for the selection process [LVZ07]

   (c) **Sequential searching method:** Sequential Method is used for the selection process [JSA09]. These methods can be used in case of equality of the replica's attributes where Euclidean distance equation can be sequentially used to find the shortest distance between the user requirement and the list of replicas.

(d) **Classification Modeling:** Used to classify the replicas using the history information to select the best one such as K-Nearest Neighbor rules [RBA04].

(e) **Predictive Modeling:** Used to predict the best replica depending on the history information such as regression [VS03] or Neural Network techniques [RBA04].

(f) **Data Base Segmentation/ Clustering:** In this method the replicas are grouped into different clusters, the replicas within a cluster have similar characteristics so the distance between two replicas in a cluster is smaller than between two replicas belonging to different clusters. These are approaches where different clustering methods used:

   i. **K-means algorithm** [JSA09]

   ii. **Genetic algorithm** [AAHJ10]

   iii. **Grey-based Rough Set Theory (GRST)**.

   In our selection techniques we used GRST with the following concepts:

   A. **Lower and Upper approximations** [AWNR10d]

   B. **Reduct** [AWNR10c]

2.8 Summary

This chapter has presented the following. First we give an introduction to Brokers in Data Grids. Then presents an introduction to submitting a job to the Broker. A comparison of four different file access patterns is presented followed by previous selection strategies. Some of them are used by well known Grid infrastructures such as Globus [gloc] and GridBus [LAB06] and others are proposed in the literature. Taxonomy of replica selection strategies is given and we related it to the meaning and the importance of the optimizer (we proposed) is discussed in the next chapter.

Chapter III

A Service Oriented Broker for Replica Selection in Data Grids

3.1 Introduction

In this chapter, the general aspects of the proposed service oriented broker along with the design outline are explained. The main objective is to design a separate scalable service oriented broker for the selection process that will locate "best" set of replica providers to move large data files. The proposed broker makes use of the core services, like the network monitoring services mechanism such as *Network Weather Service* (NWS) and *Iperf*, to improve the selection criteria depending upon dynamic network traffic conditions and improves users received quality of service with a sustainable set of data resources.

A good selection technique is a solution to many Data Grid-based applications such as *Climate data analysis* and the *Grid Physics Network* which requires responsive navigation and manipulation of large-scale datasets. As we have seen in Chapter 2 The selection process can affect the total execution time of data grids job in three ways:

1. Reducing time of the discovering the physical location(s) of the requested file
2. Reducing time of selecting the best replica provider site(s)
3. Reducing transfer time of the requested files

Usually in Data Grids, data transfer time significantly affects the total execution time. To minimize file transfer time, network links states between computing site and replica providers sites should be taken into account by the selection broker when time is one of the QoS parameters in user's requirements. The site(s) whose network link can satisfy the minimum data transfer time is (are) selected as a best replica provider site. Although, network latency and bandwidth play a major role in selecting the best replicas, other source of delay such as disk I/O plays an important role as well.

Now having seen the broad concepts of the problem we shall understand more about broker architecture in the Service Oriented Architecture B-SOA model.

3.2 Generic Broker/ Service Oriented Architecture B-SOA

In this section the general concepts of service oriented broker architecture in Data Grid, are explained. As it is shown in a Figure 3.1 a *service-oriented architecture* is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other are needed. In an B-SOA environment, there are several types of components (services) should be considered to increase the broker consolidation, these components can be grouped into: the lookup services, resource (data) filter services, resource ranker services and resource make-match services. The lookup services contain a task for collecting specific information about data providers such as their physical names and addresses. Resource filter services are used to rate the providers into desirable and less desirable providers groups. Resource ranker services is used to rank the desirable providers according to user's requirements. Resource make-match service contains a policy to perform matching between user/application requirements and the available providers characteristics. Data information services provide up to date information about data providers.

The first service-oriented architecture were Object Request Brokers (ORBs) based on the CORBA specification [sob]. For effectiveness of a service-oriented architecture, we need a clear understanding of the term 'service'. A *service* is a function that is well-defined, self-contained, and does not depend on the context or state of other services. The Service Oriented Broker takes care of all of the details involved in routing a request from client to object, and routing the response to its destination. Figure 3.2 rates a basic service-oriented architecture. It shows a service consumer at

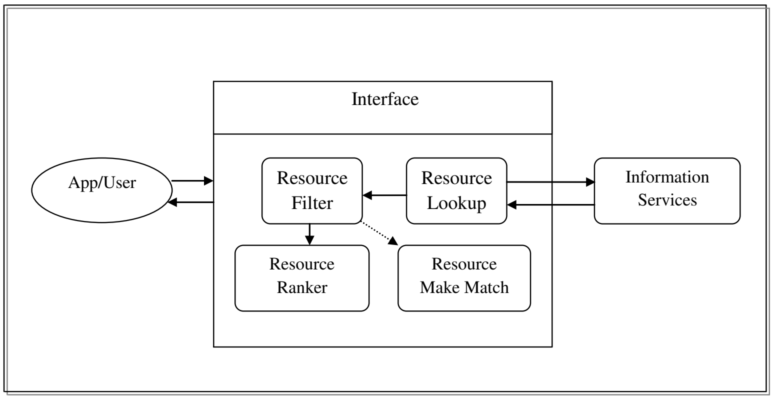

Figure 3.1: General architecture of service oriented broker in Data Grid

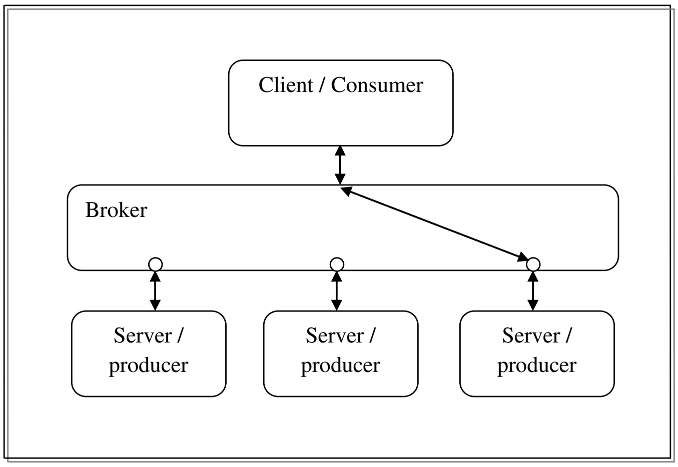

Figure 3.2: Service Oriented Architecture

the top sending a service request message to a service provider at the bottom. The service provider returns a response message to the service consumer. The request and subsequent response connections are defined in some way that is understandable to both the service consumer and service provider.

3.2.1 Some Issues and an Example

While general broker designs exit there is a need to design a new broker for replica selection problem. This is needed when the user/application wants to get data and cannot access all the attributes of that data such as, costs, quality, suppliers. Users or applications then just tell the broker their maximum price and some other requirements, and the broker starts looking for it. Here we look at certain issues that a service oriented broker would face and study an example from Globus [GII]. The

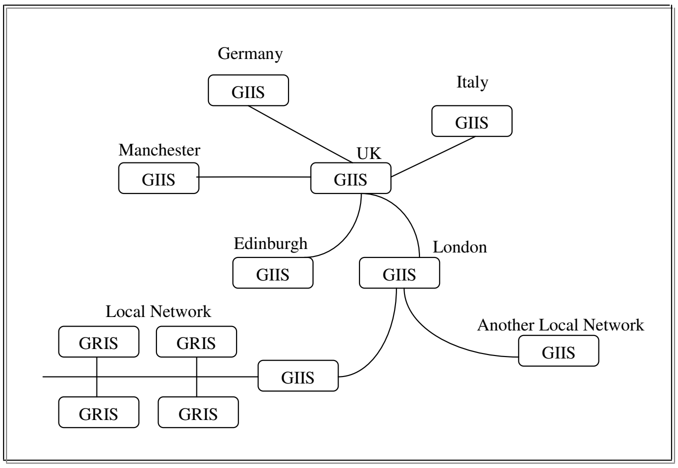

Figure 3.3: The structure of GRIS/GIIS relationships

service oriented broker would be used in the following cases:

- If the relation computing site-replica provider is not fixed because there are many suitable providers or the availability of the providers changes over time
- The choice of service provider depends on some criterion that is complex enough to be delegated to a separate component
- If the user/application wants to be independent of the physical location of the service provider (location transparency)

The client requests a specific service. It arranges its request in a specific format and sends it to its broker. The broker then selects the most suitable server to process the request. When the link between the client and the server is set up, they may start communicating directly, freeing the broker. In the following we take up an example from the GRIS/GIIS. This is shown in the Figure 3.3.

In Globus, the *Storage Request Brokers* (SRB) is an example of a service oriented broker. SRB can access data which can be used on an application wide level. In order for a consumer to find out about the network access attributes to a provider [GII], the query for this is handled by a service which provides configurable information component called as *Grid Resource Information Service* (GRIS) and a configurable aggregate directory component called a *Grid Index Information Service* (GIIS). GRIS is a central directory service, i.e. a local machine containing a grid resource information service that stores and publishes *event* or *service* information that a consumer will receive and use. The information from these local GRIS servers would then be updated on a local GIIS server that would communicate with other GIIS servers in order to supply the metric. Currently, under the Data Grid networking context, it is proposed that information should be stored on each machine on a GRIS node as shown in Figure 3.3.

After this general overview of service oriented broker architecture, the proposed service oriented broker is explained in section 3.6.

3.3 Generic use-case for the replica broker

If one considers the simplest example in a data grid, the following use-case scenario is generated. The common usage scenario of the system is a combination of three main steps:

Step 1: Job submission by the user or application. A Data Grid Job generation is done by the user via a graphical user interface, namely the user/application interface. A data grid job is a list of files that should be transferred to a specific computing element for processing to get results

Step 2: Execution of the job in the Grid, after the data grid job has been created, it is submitted to the Job manager for execution. The Job manager interacts with the replica broker service (Agent), which is responsible for intelligent matching between the user's files requests and the available data providers. After the broker has successfully matched the jobs with the proper replicas, the required data sets are moved from the selected site to computing site or cluster

Step 3: Retrieval and annotation of the results, after the execution has completed, the final results of the data mining operation are stored either in the Grid itself or on the user's client machine

In the next section job submission process is explained to give an interview about steps before Data Grid job reaches Replica Broker when the job is sent via Resource

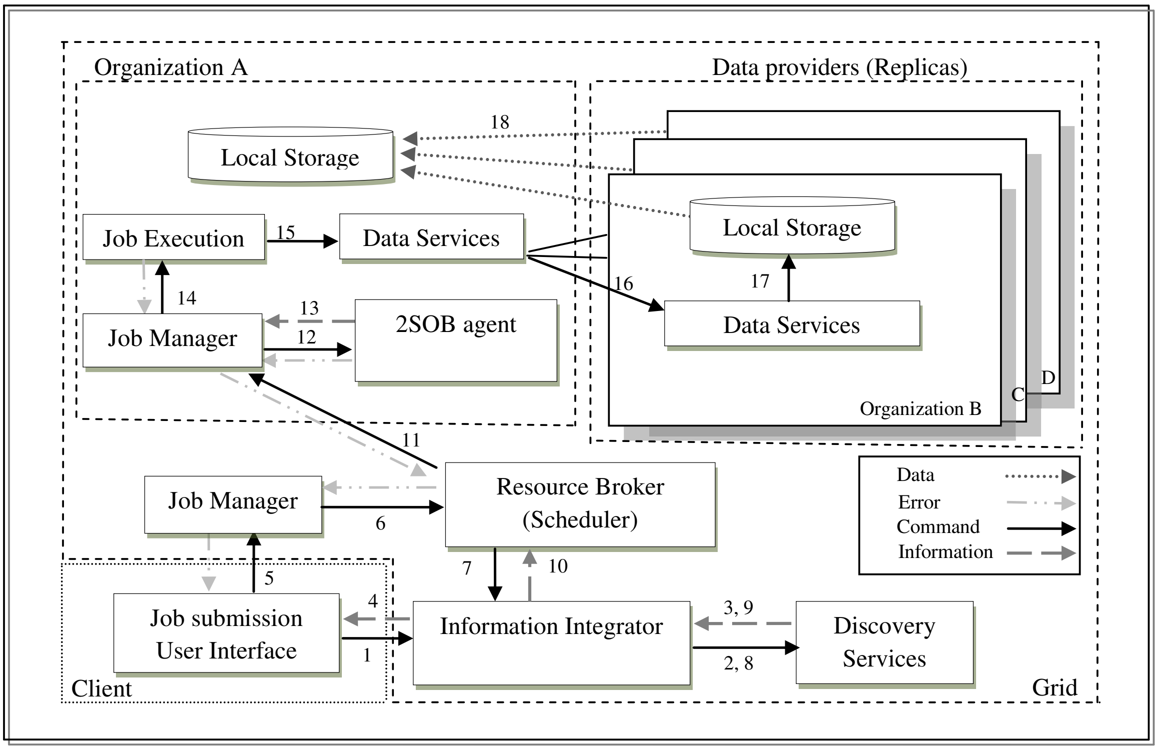

Figure 3.4: Interaction of services during job submission in the two phased replica selection system

Broker (not directly from user/application).

3.4 The job submission process

This section is to explain the sequence of steps in the process of submission a data grid job in case of our proposed broker. This is may be added to the middleware of Data Grid. Figure 3.4, illustrates the interaction of services and components during composition of the Data Grid job submission, which corresponds to steps one and two mentioned in section 1.4.1. The figure does not describe step "*three*" of the previous requirements section, retrieval and annotation of results, as this step does not depend on the replica broker.

The following steps show the sequence of the main operations of executing a data grid job as shown in Figure 3.4:

Step 1: The process of job submission starts at the user interface on the client machine

Step 2: The step of composing the Job involves querying the Information Integrator service framework for available services providers that have enough storage and computing elements

Step 3: The Information Integrator service gets all available providers of the requested services from Discovery Services unit

Step 4: Both the physical locations of the providers as well as their properties attribute values are sent to the job description of the client. While the user composes the Job, the system automatically creates a job description in the background according to the user's choices regarding the data he/she wishes to use. Thereby, by contacting the catalogue via Replica. Location Service (one of discovery services), the list of all physical locations of the required data are entered automatically into the job description without direct interaction from the user. During this process the user may specify parameters of its request. However, this job description does not specify the machines to execute the jobs on

Step 5: In case multiple storage and computing providers are available with different characteristics, then after the Job has been compiled, it is submitted to the Job manager for execution

Step 6: The Job manager resides on a dedicated machine in the Grid and is responsible for executing the Job in the correct order. During execution, the Job manager automatically passes all job descriptions to the Resource Broker (scheduler)

Step 7: The Resource Broker may send a command to the Information Integrator and asks more information related to the available providers and their attributes

Step 8: The Information Integrator sends a command asking about the required information from Discovery Services

Step 9: The Discovery Services send the information to the Information Integrator

Step 10:The information is sent to the Resource Broker for analyzing. The Resource Broker (scheduler) chooses an optimal computing site (has enough storage)based on the information about job requirements, data sizes, data permissions and workload on individual computing sites in the Grid. It retrieves this information partially from the job descriptions themselves and partially from the Information Integrator framework

Step 11: According to the chosen computing site the broker completes the job description regarding the stage-in and stage-out of input and output files. It finally passes the descriptions to the manager of the job on the selected computing site

Step 12: In the selected computing site, whenever a new job starts, the job manager sends a request to the Replica broker (2SOB) to initiate an instance of it for the particular job

Step 13: The initiated 2SOB, is another broker, it is to choose best set of replica provider sites which are stable and have a close match to requirements of the user. The list of replica providers with their attribute values are retrieved from the discovery services such as RLS and NMS. According to the chosen replica sites, the optimizer completes the job description regarding the stage-in and stage-out of input and output files and sends the information to the Job Manager of the computing site (the proposed 2SOB broker is explained in the next section)

Step 14: Job manager finally passes the best set of replica providers to the job execution service

Step 15: The job execution service contacts the data service

Step 16: Data service of the computing site communicates with data services in the selected sites of the best set . Data transport service such as GridFTP is used to transfer the required data files from the selected replicas sites

Step 17: Data services of the best replica providers contact their local storage via Data Transport Services. It is responsible to send a command to the local storage to prepare required data

Step 18: The required data is sent simultaneously from the selected replica providers into the computing site (the one which was selected by the scheduler)

Step 19: Both the physical location of the data as well as the results of the operations performed are transparent to the user

Step 20: Release resources

3.5 Requirements of brokers in Data Grids

A replica broker in Data Grid must address the following requirements:

1. The broker has an automated selection of the most suitable replica providers from the list of replicas available at the requesting time
2. The broker has to carry out this operation without direct user interaction [KS06]

The both requirements have been taken into account during designing our approach, 2SOB.

3.6 Two-phased Service Oriented Broker architecture 2SOB

As it is mentioned before, the proposed broker is more efficient when there is a need to retrieve different segments of data from multiple data sources. The Service Oriented Broker does the file transfer in three cases:

1. Broker receives a request from Gird user/application.
2. Scheduler (broker) has been assigned data to perform efficient replication at various grid sites.
3. Huge data has been generated and should be distributed in different Grid sites.

In the first and second cases the broker will select the best set of providers to retrieve the required data. In the third case the broker decides best storage for saving the generated data. In all the cases the latency of the links is the first issue to be taken care of, comparing to the other user/application requirements.

The main objective behind proposing a Service Oriented Broker is, to select set of *best* replica providers set. This practically means to select the optimal number of sites with good latency (the amount of time taken by a packet to travel from source to destination), and to maintain good bandwidth to speed up and fully utilize the capacity (throughput) of a network [LAB06]. An Intelligent Optimizer is the main component in the proposed broker, 2SOB architecture. Following are the two phases of 2SOB:

1. **Phase One**: It is known as Coarse-grain phase. It uses *Efficient Set Technique(EST)* to sift replica sites that have low latency (uncongested links). This phase generates two sets, first Maximum Frequent Set (MFS), which is known as *Associated Replicas Set* (ARS). ARS represents maximum number of replica providers with stable network link property. The second set is called *Frequent ItemSets (FISs)* which is used to generate associated replicas rules as shown in Figure 3.5.

2. **Phase Two**: It is known as Fine-grain phase. It uses *Grey-based Rough Set Theory* to extract *Best Replica Sites* (BRS), number of providers that have close match to the user requirements. BRS is used to generate association rules in *Associated Replicas Rule Repository* (AR^3). Generally the AR^3 is useful when the files requests are related. It means that the required files belong to a single collection.

Traditionally, a centralized approach to resource management is used in the resource brokers, wherein a single node is responsible for decision making. An example of such an environment is the Condor [LLM88]. Obvious disadvantages to this approach are scalability and a single point of failure.

In Condor an efficient recovery mechanism is used to address failure and has been proven to scale to thousands of resources and users. Even though, there is a more fundamental problem with this centralized approach when applied to Grids.

In these highly distributed environments, there are numerous user communities and shared resources, each with distinct security requirements. No single resource broker is likely to be trusted by all of these communities and resources with the necessary information to make decisions. At the extreme, each user may need his

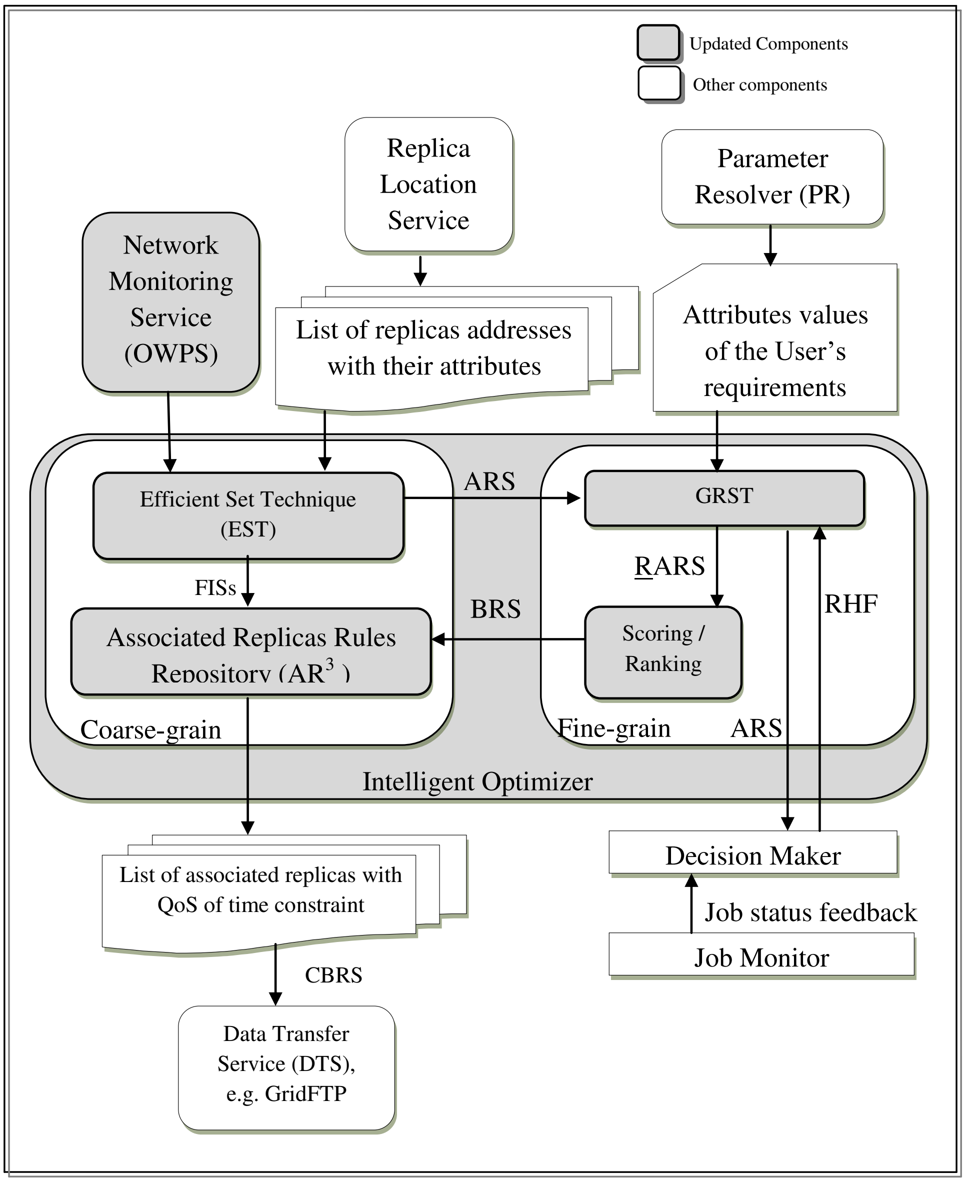

Figure 3.5: Architecture of the Intelligent Optimizer in Two phased Service Oriented Broker

or her own broker, because only that user has the authorization to gather all of the information necessary to make brokering decisions. For this reason, we have designed a decentralized selection brokering strategy wherein every client who requires access to a replica, performs the selection process rather than a central manager performing matches against clients and replicas as shown in Figure 3.6.

Another reason is, if the huge number of users and resources which may exist in the grid are considered, then using centralized approach may cause a bottleneck to the replica management system as a centralized Agent targeted by all user/application requests. The solution to this problem could be by replicating (multi-threading) the 2SOB into several Data Grid nodes to reduce the load on the centralized Agent. So, the job can be processed in parallel in order to scale up numerous requests at a time. That means, using 2SOB, Data Grid architecture provides a scalable infrastructure for the management of replica selection jobs and data that are distributed across Grid environments.

2SOB is a middleware responsible for carrying out a set of tasks related to Data Grid user job submission. These tasks include interacting with Replica Catalog (RC) to resolve logical data set names, finding preliminary set of replica provider sites for data transfer, job submission and management by interacting with other sub-systems such as the Job Submission Service (JSS).

The replica broker is a background process known as listener or daemon process works in a server that boots with a head process named "2SOB Listener process". 2SOB Listener process listens to a TCP/IP socket and wait for client requests as shown in Figure 3.6.

This process is responsible for capturing the user/application requests and forking a new process known as "2SOB Agent process" to serve the incoming requests concurrently. That means, upon client interaction, a thread is spawned to handle the client messages using a new available port.

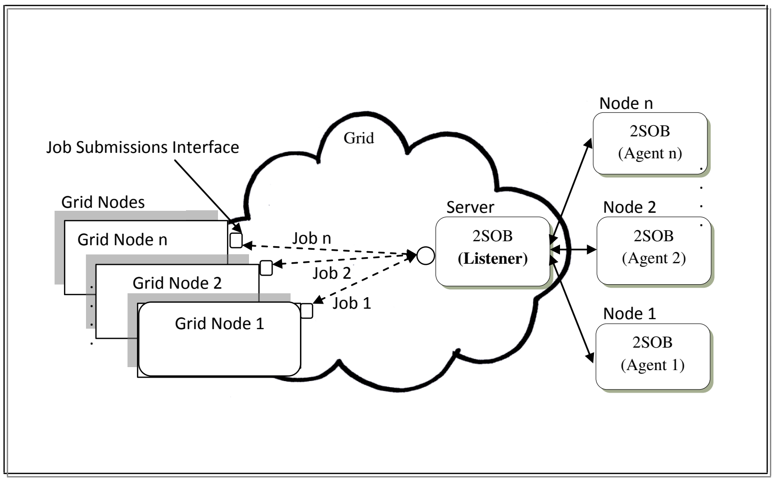

Figure 3.6: Replica Broker infrastructure

The 2SOB master daemon acts as the preliminary broker for Agent thread to handle individual client. The Agent process acts as the hub for carrying out tasks by communicating with the JSS, RC and the local Job Registry (JR) database through the Data Base Management System (DBMS) interface to ensure job state consistency and persistence of queued jobs. This section shows the main steps of replica brokering algorithm which are used in 2SOB for replica selection process. The actions of the daemon are described as follows:

- Input: One or more job requests
- Action: Initiate an instance of 2SOB (Agent) and assign to each job
- Output: Each job is assigned to a single Agent

The actions of the Agent are described as:

- Input: Single Data Grid job request with a time constraint for the job execution
- Action: Select the most appropriate replica provider(s) to transfer the required data
- Output: Set of Best associated Replicas Set (BRS) that have a close match to the user's requirements

Actually, the proposed broker (2SOB) is useful when there is a need to retrieve different segments of data from multiple data sources as it is shown in Figure 3.7.

Algorithm 3.1 represents an outline pseudo code of typical 2SOB server that generates concurrent 2SOB Agents. When a new request comes, "accept" method returns. Then the server calls "clone" to create a child process named "2SOB Agent Process". The Agent Process serves the incoming request while the parent process, 2SOB Listener Process waits for another request to come. Creating a new process for every data grid job might be expensive in terms of memory consumption as every process has its own separated segments in memory.

We use "clone" rather than "fork" as the first creates a "light weight" process which shares parts of its execution context with the parent process, such as the memory space.

Algorithm 3.1 Outline pseudo code for typical 2SOB server Algorithm

```
Initialize:
   [1] pid_t pid
   [2] int 2SOB-listenerId, 2SOB-AgentId
Input: A Data Grid job (J1)./*J contains name of required files*/
Output: Set of Best associated Replicas Set (BRS) that have a close match to the user's
requirements
Begin
   2SOB-listenerId = socket(...) /* Assign 2SOB Listener Process with well-known port */
   bind(2SOB-listenerId, ...) /* Bind a resources to the new child such as port number*/
   listen(2SOB-listenerId, 1)
   while(true) /* Infinite loop */
     begin
      2SOB-AgentId = accept(2SOB-listenerId, ...)
      if( (pid = fork()) == 0) then
        begin
          close(2SOB-listenerId) /* Child process (2SOB-AgentId) closes the parent process'
socket(2SOB-listenerId) */
        BRS = ARD(2SOB-AgentId) /* Serve the user request by calling ARD strategy and
return the Best Replicas Set (BRS) */
        close(2SOB-AgentId) /* User request is served and the resources are released */
        exit(0) /* Child Termination*/
        end if
     end while
   End
```

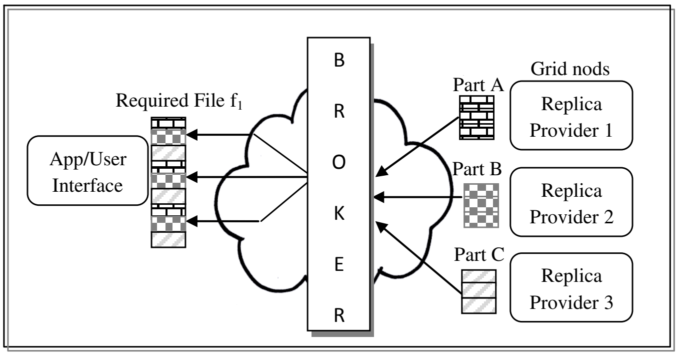

Figure 3.7: Concurrent bulk data transfer of file (f_1) from multiple providers

Figure 3.8 shows the behavior of 2SOB and the related components which are:

1. **Resource Requirement Interpreter**: It is used to interpret the requirements of the user/application

2. **Replica Discovery Unit**: It is used to communicate with the Dynamic Grid Resource Information Services (DGRIS) to obtain a list of available replica providers by contacting GIIS server(s). Also, each resource's GRIS is contacted for static and dynamic resource information (hardware and software characteristics, current queue and load, etc.)

3. **Replica Management Unit**: It is used to receive network reports and list of replica providers with their attribute values. This unit generates the NHF to Coarse-grain phase and also generates standardized table of replica providers attributes to be used by Fine-grain phase

4. **Associated Resource Discovery Unit**: It is used to sift the set of replica providers which have uncongested network links that are called Associated Replica Sites (ARS)

5. **Ranking and Filtering Unit**: It is used to rank the associated replicas set to select Best Replica Sites (BRS)

6. **Decision Maker Unit**: It is used to save all expressions of the decision makers committee preferences on replica providers or on attributes of replica providers in *Reputation History File* (RHF) (RHF is be used to help ranking the replica providers or selecting the most desirable replicas, as explained in Chapter 5)

7. **Resource Reservation Unit**: It is used to reserve the resources of BRS and also to release any reservation made to unselected replicas

8. **Data Transport Service Unit**: It is used to transfer the replica from the selected replica site to computing sites

A User/Application on the Data Grid submits a connection request to a "2SOB Listener process" that resides on the destination node in the cluster. A 2SOB listener process is a process that manages in-coming connection requests. 2SOB has two phases which are briefly explained in the next two chapters.

As it is shown in Figure 3.5, the Optimizer communicates with different Data Grid services explained in the following subsections:

3.6.1 Network Monitoring Service (NMS)

The proposed broker, Two phased Service Oriented Broker utilizes network monitoring service; a core service to get a real picture of the links latency. Currently, in Data Grid infrastructure, the network monitoring service measures the latency using *Round Trip Time (RTT)*. With round trip-based measurements, it is hard to isolate the direction in which congestion is experienced. Actually, in bulk data transfer the single trip from provider Replica Site to the requester Computing Site is required to be monitored. Therefore in our proposed broker, One Way Ping Service OWPS [owa] is suggested to be used to get *Single Trip Time (STT)*. STT is a time taken by the small packet to travel from Replica provider Sites(RS) to the Computing Site (CS). The STT delays include five types of delays which are: packet-transmission from RS to CS delays, the transmission rate out of each router and out of the replica site, packet-propagation delays (the propagation on each link), packet-queuing delays in intermediate routers and switches, and packet-processing delays (the processing delay at each router and at the replica site) for a single trip starting from Requester Site (RS) to the Computing Site (CS).

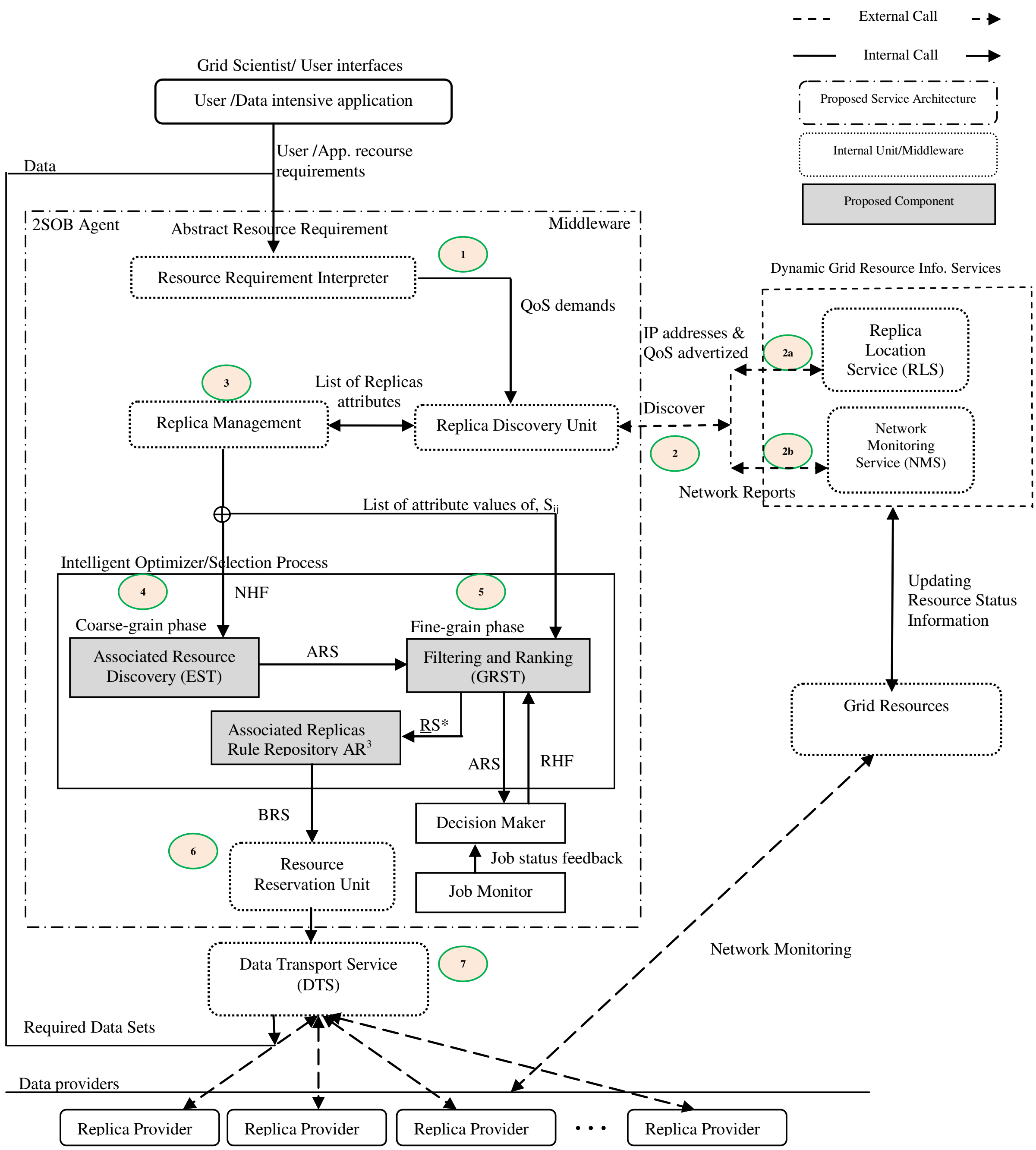

Figure 3.8: 2SOB architecture

3.6.2 Replica Location service (RLS)

RLS is the service that keeps track of where replicas exist on physical storage systems. It is responsible for maintaining a catalog of files registered by the users or services when the files are created. Later, users or services query RLS servers to find physical locations of replicas.

3.6.3 Data Transport Service (DTS)

As it is mentioned in Chapter 1, the standard tool for data transfer between the Gird sites or for downloading from Grid sites is *GridFTP* [gloc]. GridFTP is a high- performance, secure, reliable and multi streams, included in most of the storage management systems. It uses TCP as the transport layer protocol. In our work we use a data grid simulator called OptorSim. The next section presents the simulator OptorSim in brief, its architecture and the execution flows of the replication and selection algorithms in the simulator.

3.7 Simulation using OptorSim Simulator

To evaluate our approach a simulation package called *OptorSim* is used [BCC+03]. OptorSim is a Data Grid simulator designed to allow experiments and evaluations of various replication optimization strategies in Grid environments

3.7.1 Architecture

The simulator design assumes that the Grid consists of a number of sites, each consisting of zero or more computing elements and zero or more storage elements. The computing elements provide computational resource and the storage elements serve as data storage resources for submitted jobs. A resource broker acts as a meta-scheduler that controls job scheduling for different computing elements. A job in OptorSim must access a set of files which may be located locally or at different storage sites.

3.7.2 Grid configuration Files in OptorSim

Using OptorSim simulator our program gets input from the three configuration files:

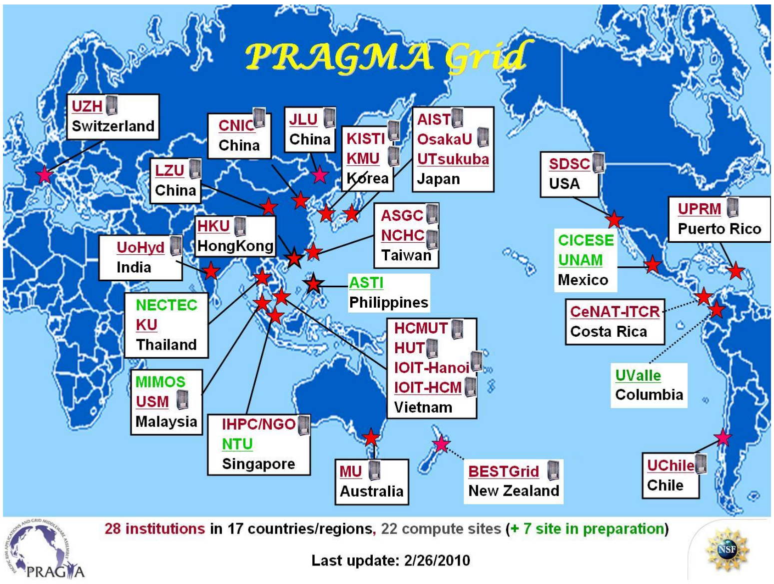

Figure 3.9: PRAGMA GRID

3.7.2.1 The Grid Configuration File

In the simulator the grid configuration file is used to describe the topology of the participating sites, their associated network geometry and the content of each site shows resource availability. Being a node of an intra-grid we encode the real life example of the PRAGMA Grid into the simulator (see figure 3.9). Therefore our grid configuration file reflects the real network nodes and network links conditions of the PRAGMA Grid [pra].

In the simulation, we distributed the files in 19 sites which are denoted as: $U = \{S_1, \cdots, S_{19}\}$. S_0 is the computing site that has enough computing elements and storage to store the requested files for jobs represented by (venus.uohyd.ernet.in). J is list of five jobs submitted to computing site where, $J = \{J_1, J_2, J_3, J_4, J_5\}$. The required files of J saved in 9 sites which are: $I = \{S_1, S_3, S_4, S_5, S_6, S_7, S_8, S_{14}, S_{18}\}$ as shown in Figure 3.10.

Each link between two sites shows the available network bandwidth which is expressed in Mbits/sec (M) or Gbits/sec (G). The circles referred to the sites which are separated by stars referred to routers.

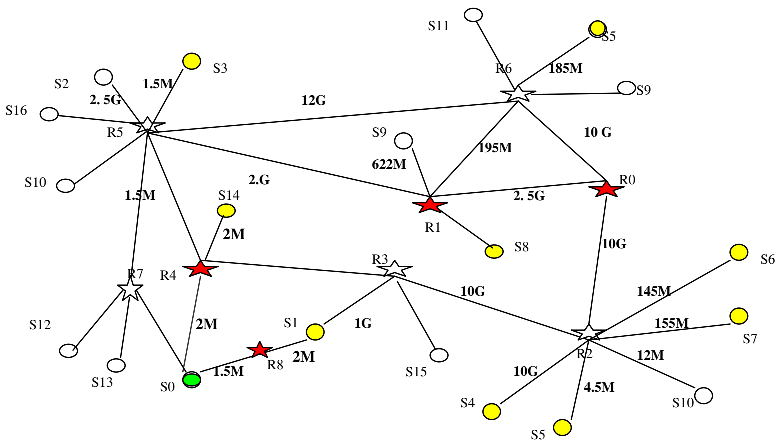

Figure 3.10: PRAGMA Grid and their associated network geometry

3.7.2.2 Job Configuration File

This file contains names of jobs, list of required files for each job, a list of logical file names, their sizes in MB/GB and their unique numerical identifiers. It also contains job selection probability and schedule table with job execution time for each computing element.

3.7.2.3 Background Network Configuration File

This file is used to describe background network traffic, it is a site-by-site matrix, having for each pair of sites the name of data file containing the relevant STT information, where in our simulation STT is $RTT/2$, the Mean of STT and the Standard Deviation; keeping the source sites in the rows, and the destination sites in the columns. The configuration file initializes different parameters for running the simulation. These parameters may include information such as total number of jobs to be run, file processing time, delays between each job submission, maximum queue size in each computing element, file access pattern, the optimization algorithm used, etc.

As mentioned above, a job will typically request a set of logical file name(s) for data access. The order in which the files are requested is determined by the access pattern. In this simulation, we experimented with four access patterns: sequential (files are accessed in the order that has been stated in the job configuration file), random (files are accessed using flat random distribution), unitary random (file requests are one element away from previous file requests, but the direction will be random), and Gaussian random walk (files are accessed using a Gaussian distribution) as it is explained in Chapter 1.

During the simulation, some drawbacks of previous selection strategies have been noticed, the next section is to explain them.

3.7.3 Other selection methods

This section is to explain some of previous selection methods that are used in resource broker. Mathlab tools are used to simulate neural network and regression models. After that the results are used in the network file and job configuration file of OptorSim. The next section presents the simulation results and analyzes the performance of the replica selection algorithms with different performance metrics.

3.7.4 Neural Network and Regression Models

The assumptions of the previous methods [RBA04, VS03] are used to estimate the transfer time between sites by neural network, we use a neural network with 3 layers. The input layer consists of 3 neurons, and the hidden layer consists of 2 neurons. In the output layer, there is only one neuron used to predict the current transfer time between two sites.

The backpropagation algorithm is used for training the NN. When a file transfer has been made between two sites, the file size, the available network bandwidth and transfer time are written, which is later used for training and testing the neural network. To train a neural network data should be between zero and one so, we normalize the data first by dividing through a large value to make data in the [0, 1] range.

Each training set consists of three values; namely, file size, available network bandwidth experienced between sites to transfer the file, time elapsed to transfer the file, e.g., transportation time for the file. We start training the NN with the first 20 data sets, called the training data sets.

Each set is presented to the neurons of the input layer and the whole network is trained with the backpropagation algorithm so that the output neuron can predict the transfer time between sites. After training the neural network with first twenty data sets, more values are included in the training set and the prediction is made. Weights are adjusted in the neural network so that the prediction error will be minimized.

We train the NN for 3000 epochs with different learning rates and momentum values but we get the optimal error for the learning rate value 0.25 and momentum of 0.08. Learning rates in neural networks determine how much we change the weights in each step. If the learning rate is very small, the network will take a lot of time to converge. On the other hand, if the learning rate.

After 20 file transfers, the information like, size of file and bandwidth between sites is used as a training data and presented to the input layer of the neural network. Then the trained network is used to predict the transfer time between sites that hold replica of the file for 21^{st} file transfer request. The site with the lowest predicted file transfer time is considered as the best replica provider.

The difference between time of getting the file from predicted site and time of getting the file using traditional method is recorded in the file which is used later to compare the accuracy between the neural network prediction technique and regression based prediction technique and traditional method. In their research, they use a traditional network model that consider network as a static and unmanaged resource. However, in Grids the network is dynamic, configurable and closely monitored resource. They have not worked and applied their model to a deterministic beyond the scope of their work. So, as it is seen in the figures the prediction is not stable and depends on the access pattern of the required files. This unstable results would increase the total transfer time as the Data Grid jobs are varying with the access pattens.

3.7.5 K-Nearest Neighbor Model

We tested K-nearest rule with two options, i.e., taking account replica allocation or reallocation into replica selection. The first option does not consider the scenario that replica placement or reallocation (replica moving from a site to another) is done by the resource broker. The second option considers replica allocation or reallocation decision from the resource broker. This is due to the dynamic nature of the Grid, the replicas are reallocated to different sites.

For the first one, after 30 initial file transfers, the simulation switched from the traditional model to the K-nearest algorithm model for replica selection. For the second one, the transition from the traditional model to the K-nearest model happens when first static placement is decided by the resource broker.

After replica placement is done, the sites wait for twenty more jobs to be submitted by the resource broker before a site can use the K-nearest rule. The site also uses the same method in case of relocation done by the resource broker. For both options, the number of right and wrong classifications made by the K-nearest rule is recorded. The Grid environment is dynamic where user requests and network latencies vary significantly, therefore the site selected by the K-nearest rule may not be best site for replica selection for current condition. Due to sudden change in network conditions, the K-nearest rule may give wrong classifications. If the K-nearest rule gave five consecutive wrong classifications the simulator switched to traditional model again. The K-nearest rule resumes its execution when a correct classification is again been made.

We use $k = 5$ for the K-nearest rule. Figures 3.16, 3.15, depict the number of right and wrong classifications for different file access patterns in the predicted replica provider S_5.

Figure 3.16 depicts the right and wrong classifications when replica placement decision is not taken account by the KNN. If the KNN prediction model returns the

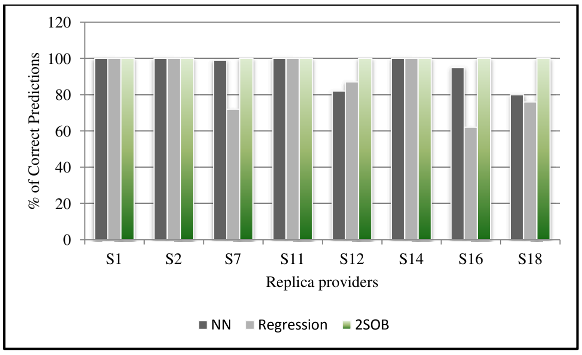

Figure 3.11: Percentage of the Correct Predictions for Sequential Access Pattern

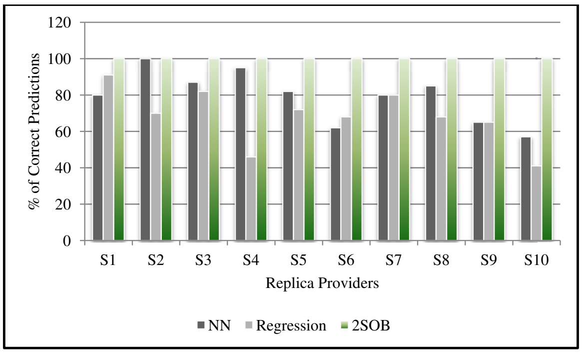

Figure 3.12: Percentage of Correct Predictions for Gaussian Access Pattern

same site as returned by the traditional replica model, the file transfer is classified as right classification; otherwise it is classified as wrong classification. Figure 3.15 depicts the number of right and wrong classifications by the KNN model when the site considers the replica placement decision.

3.8 Results Analysis

In this section we want to explain the ability of 2SOB to solve *Availability problem*, one of previous work drawbacks. For analyzing results we do comparison between the proposed broker, 2SOB and NN with Regression first and then 2SOB and KNN.

Figures 3.11, 3.12, 3.14, and 3.13, depict the percentage (%) of the correct predictions by the neural network and regression models for different access patterns. If the predicted site is the same site considered for the current file transfer for the current network conditions it is considered as correct prediction. Otherwise, it is

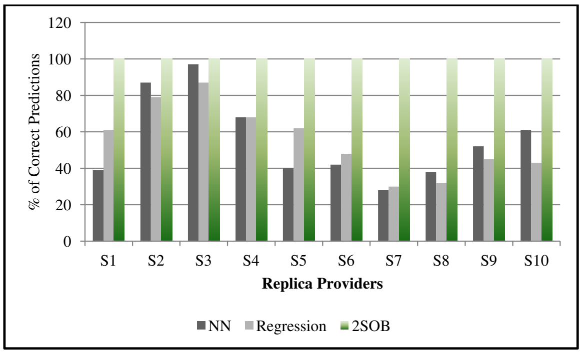

Figure 3.13: Percentage of Correct Predictions for Unitary Access Pattern

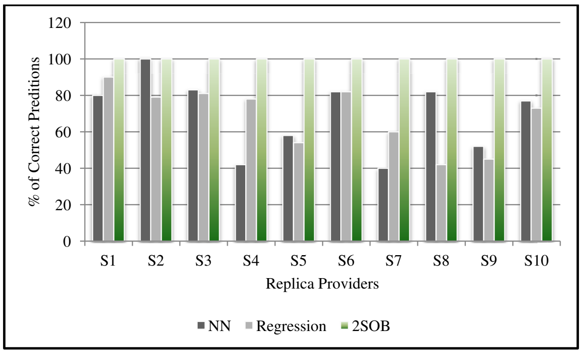

Figure 3.14: Percentage of Correct Predictions for Random Access Pattern

considered as incorrect prediction. The % of correct predictions is calculated by the number of correct predictions divided by the number of total predictions.

For example, (S_{17}) made total 174 file transfers. For 137 transfers, NN predicts the same site that would be selected for current network conditions using traditional method. NN predicts different sites for the other 37 file transfer. The % of correct predictions for neural network for sequential access patter is therefore, 78.73%.

Comparison between the performance of the proposed approach, 2SOB and KNN in both cases, with and without replica placement or reallocation decision.

Figure 3.15, and 3.16 depict that even though the number of right classification in case of placement decision has taken in account of KNN but, the total right classifications for a single replica provider site S_5 is less than number of right classifications of 2SOB with all different file access patterns.

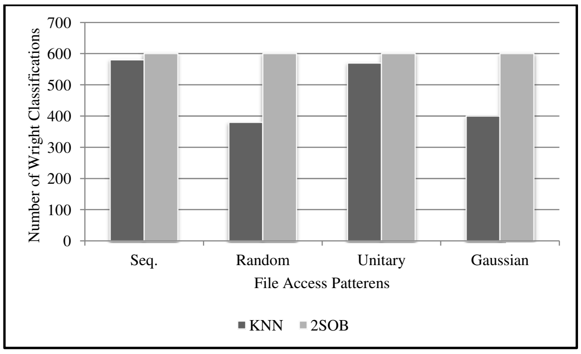

Figure 3.15: Comparison between KNN and 2SOB with Number of Right classifications of different file access patterns with placement decision

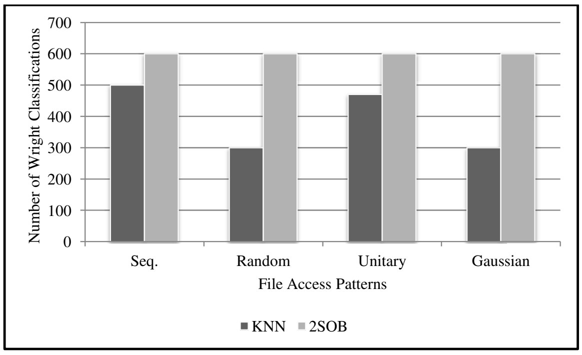

Figure 3.16: Comparison between KNN and 2SOB with Number of Right classifications of different file access patterns without placement decision

The reason behind that is, 2SOB works on the list of up to date of available replica providers whereas, KNN works on outdated data and depends its selection on the previous file transfer requests. In that case the selected provider may or may not have the required file.

The results of the simulation is, the availability of required data in the replica provider which has been selected using 2SOB is 100% while it is varies for other selection methods depending on two factors, type of file access and the method placement decision.

The availability percentage is about 15% – 50% for Regression model and about 40% – 70% for NN model and about 50% – 80% for KNN.

in the next chapter a real data of network monitoring reports from National Accelerator Laboratory called *SLAC* [slaa] are used to test and compare the selection models with our Coarse-grain model with a one type of access pattern and without reallocation decision.

3.9 Chapter Summary

In this chapter we have started with some essential concepts related to services in data grids. Then we have presented a general service oriented broker. This is followed by the example of a storage request broker in Globus. We take up the actions to be taken by a broker daemon for the replica selection. A generic use case is presented for such a broker. An internal design for a replica broker is then proposed. This is followed by a detailed description of how jobs are submitted to to such a broker. The architecture design of the proposed two phase service oriented broker is shown along with a description of its active components. Then we ended with an implementation using a real grid (PRAGMA) infrastructure and compare our approach with others.

We will take up further details of the components described here in the forthcoming chapters. Chapter 4 used to give an introduction of data mining and the association rules concepts that are necessary to explain the Coarse-grain phase then. It gives details of this concept and how it is used in the broker to improve the selection process. In Chapter 5 an introduction about Rough Set theory and its concepts is presented. These are necessary to explain the Fine-grain phase with and without decision attribute. The strategy of the fine-grain phase is applied in the Cloud environment is presented in Chapter 6.

Chapter IV

Coarse-grain phase: Data Mining-based Associated Replicas Discovery (DM-ARD)

4.1 Introduction

In this chapter we shall apply the powerful methods for discovering knowledge derived from data mining techniques to the task of replica selection problem in Data Grids. We explain the tools and algorithms used to discover the latent relationships of network attributes between a computing site called as a *requester* and other replica sites *providers*. More precisely we attempt to discover the hidden relationships of the instantaneous network latencies between a requester and the providers. Each requester is expected to use a service oriented broker that adopts the selection strategy as explained here. We aim to develop association rules here so that for example a general query like "which items are most frequently having stable latency within X period of time " may be answered by a broker.

Here we aim to apply an Association Rule Mining (ARM) technique ARM, such as the *Apriori algorithm* for mining association rules using the concept of *Maximal Frequent Sets* (MFS).The terms *MFS* and *ARS* are used interchangeably. Before explaining in detail a Coarse-grain approach it might be convenient, rather essential to review some of preliminary concepts about association Rules Mining (ARM) technique.

4.2 Preliminary Concepts

In this section the concepts of Data Mining and Data Grid with the common convergence between them are explained.

4.2.1 Notation in Data Mining (DM)

This section presents a methodology known as association rule mining, useful for discovering interesting relationships hidden in huge data sets. Association rules have received lots of attention in DM due to their many applications in marketing, advertising, inventory control, and many other areas. Association Rules can be derived using supervised and unsupervised processes [Joe09].

Following the definitions by Agrawal et al [ATS93], the problem of association rule mining is explained in the following.

- Data base (D): let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called items. Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in I.

- An Association Rule (AR), A rule is defined as an implication of the form, $X \to Y$, where $X, Y \subseteq I$, and, $X \cap Y = \emptyset$, The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively.

  To illustrate the concepts, we use a small example from the supermarket domain. The set of items is $I = \{milk, bread, butter, beer\}$ and a small database containing the items (1 indicates presence and 0 absence of an item in a transaction) is shown in the table 4.1. An example rule for the supermarket could be $\{\text{butter}, \text{bread}\} \to \{\text{milk}\}$ meaning that if butter and bread is bought, customers also buy milk.

this example is an extremely small and trivial example. In practical applications, a rule needs a support of several hundred transactions before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best known constraints are

Transaction ID	milk	bread	butter	beer
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Table 4.1: Example data base with 4 items and 5 transactions

minimum thresholds on support and confidence.

- The support supp(X) of an itemset X is defined as the proportion of transactions in the data set which contain the itemset. In the example database, the itemset milk,bread,butter has a support of $1/5 = 0.2$ since it occurs in 20% of all transactions (1 out of 5 transactions).

- The confidence of a rule is defined $\mathrm{c}(X \rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X)$. For example, the rule $\{\mathrm{milk, bread}\} \rightarrow \{\mathrm{butter}\}$ has a confidence of 0.2 / 0.4 = 0.5 in the database, which means that for 50% of the transactions containing milk and bread the rule is correct. Confidence can be interpreted as an estimate of the probability $P(Y|X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS [AS96, WHR98]

- The Improvement (lift) of a rule is defined as $\mathrm{lift}(X \rightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(Y) \times \mathrm{supp}(X)}$ or the ratio of the observed support to that expected if X and Y were independent.

  The rule $\{\mathrm{milk, bread}\} \rightarrow \{\mathrm{butter}\}$ has a lift of $\frac{0.2}{0.4 \times 0.4} = 1.25$

- The conviction of a rule is defined as $\mathrm{conv}(X \rightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{c}(X \rightarrow Y)}$. The rule $\{\mathrm{milk, bread}\} \rightarrow \{\mathrm{butter}\}$ has a conviction of $\frac{1-0.4}{1-0.5} = 1.2$, and can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions. In this example, the conviction value of 1.2 shows that the rule $\{\mathrm{milk, bread}\} \rightarrow \{\mathrm{butter}\}$ would be incorrect 20% more often (1.2 times as often) if the association between X and Y was purely random chance

- A Frequent Item Set(FIS): An itemset X is said to be a frequent Item Set (FIS) in T with respect to $(min - supp)$, iff $\sigma(X) \geq min - supp$ [Puj01]
- A Maximal Frequent Set (MFS): A FIS is called Maximal Frequent Set (MFS) if no supper itemset of this itemset is a FIS

4.2.2 Notation from Data Grids

Here briefly we review some notation from Data Grids that is being used to explain the concepts.

1. Replica provider Sites or Replicas (RS): It is a grid site(s) which has a storage element which is used to store copy of some files in short it is called Replicas. RS or S_i, where, $i = \{1, 2, \cdots, M\}$, and M is the maximum number of replicas. each replica provider site Si has a vector of attributes determined by the administrator of the site which is denoted by: $S_{i,j}$, where, $j = \{1, 2, \cdots, N\}$, and N represents the maximum number of attributes for each replica
2. Replica: a copy of requested file (f) each replica's site S_i has a vector of attributes determined by the administrator of the replica provider.
3. User Request (UR): is a vector of attributes denoted by
   a_h, where, $h = \{1, 2, \cdots, Q\}$, and Q is the subset of user attributes and $(Q \leq N)$
4. Network History File (NHF): It is a file that contains a complete history of values of Round Trip Time (RTT) or Single Trip Time (STT) of connected replicas at interval time $[t_0 - t_z]$. This file can be formed using a monitoring tool such as PingER [pin], for example NHF contains:
   - Rows = Transactions (T) within an interval time $[t_0 - t_z]$
   - Columns = Identification of Replica Sites $(IDRS)$, S_i

5. Session: let NHF be divided into s sessions (partitions) of slots of time, $P = \{P_1, P_2, \cdots, P_s\}$, where P_s is called the latest session of NHF. It is the most recent session (partition) within the time interval $[t_{z-l} - t_z]$, and l is the size of interval of P_s in NHF which is denoted by $NHF[P_s]$

6. Data Gird Job (J): A job contains set of data files need to be accessed and analyzed, for example, $J = \{f_1, f_2, \cdots, f_t\}$, where, t is the total number of files

The above mentioned concepts of data mining are applied on the information from the Data grid to mine the data about the links which connect replicas. To the best of knowlege, the concept of Data Grid Mining is used first time is this context. This novel idea is made more elaborate in the next section where the parallels between Data Mining concepts and Data Grid concepts are drawn out.

4.2.3 Parallels between Data Grids and Data Mining

Before explaining the present approach it might be convenient, rather essential to review some concepts about the parallelism between data grid and data mining techniques. Here we synthesize the novel concept of Data Grid Mining, a type of data mining which may be generally useful in assessment and management of data grid systems. In this work we use it in a very specific instance that is to find out relationships between the instantaneous network conditions between replicas. It takes advantage of the huge amount of information gathered by a network monitoring service to look for patterns in replicas' link behavior. The parallelism between Data Mining and Data Grid Mining concepts is sketched out in the following aspects:

- *Item*: An item in Data Mining is an attribute value which represents the presence or absence of an attribute in a specific transaction; while in Data Grid Mining it is a Boolean value that determines whether or not the link is stable in a specific transaction
- *Data Base (D)*: the Network History File (NHF) in Data Grid Mining represents the Data base of Data Mining
- Transactions (T): Each transaction in data mining is a set of items (attributes), $T = \{i_1, i_2, \ldots, i_n\}$ where n is number of attributes. In case of Data Grid, Each transaction (T_k) is a set of times in milliseconds of either single or round trips from replica providers to the computing site T_k=$\{RTT_1, RTT_2, \cdots, RTT_N\}$, where k is a slot time $0 \leqslant k \leqslant z$, and N is the number of replicas (providers)

- Association Rules (AR): The purpose of mining association rules in a database is to discover hidden information between attributes. In Data Grid Mining the transferring rules (TR), the rules represent the most likely relationships among links connecting replicas and the power of the cooperating among the replica provider sites are represented by CTR set
- Frequent Item Set (FIS): Tn Data Mining it is represented by set of associated objects ARS [Puj01]. In Data Grid Mining it represents the set of replica provider sites located in the different locations with similar behavior of their links (uncongested) it is known as, Maximum Frequent ItemSet MFS

Table 5.2 shows the notation of all symbols being used in the algorithms.

4.3 Coarse-grain phase

As we mentioned before, a Coarse-grain phase uses association rules concept upon data sets of the Network History File [AWNR10a]. The association rules algorithm (Algorithm 4.1) is the process of finding hidden relationships in data sets and summarizing these patterns in models. It is meant for combing through data to identify patterns and establish relationships that allow an applicant to discover knowledge that is hidden in large data sets. Association is looking for patterns where one event is connected to another event. In our model, coarse-grain phase is when the problem of discovering the association rules in Data Grids can be decomposed into two subproblems:

1. First is finding Frequent Item Sets (FIS). The Maximum Frequent Set (MFS) in our context is called as an Associated Replicas Sites (ARS). This can be done using Efficient Set Technique (EST) [AWNR10a]
2. Second is deriving the Transferring Rules (TR) from the frequent item sets (FIS) [AWN+10a]. Figure 4.7 shows the output of Coarse-grain phase and Table 5.2 lists the notation used to write the algorithms of Coarse-grain phase

Both outputs (ARS and TR) represent of Coarse-grain phase as shown in Algorithm 4.1.

No.	*Symbols*	Description
1	STT	Single Trip Time
2	RTT	Round Trip Time
3	FIS	frequent item set
4	MFS	Maximum-Frequent-Set
5	MR	Maximum number of items in a FIS or rule restricted by a replica broker
6	T	Set of transactions of $STTs$ or RTT of an interval time $[t_0$-$t_z]$
7	AR	Associated-Rules
8	ARS	Associated Replica Sites, ARS_d , where $d = 1, 2, \cdots , m$, and m represents number of associated sites and $m \leq M$
9	$k - itemset$	An itemset with k items
10	C_k	set of candidate of $k - itemsets$
11	L_k	Set of large k-item sets (set of itemsets have minimum support)
12	l_i, l_j	Any of the frequent $(k - 1) - itemsets$ within L_{k-1}
13	$l_i[m]$	It is a $m - th$ item in itemset l_i
14	$min - conf$	Minimum confidence value
15	$min - supp$	Minimum support value
16	NR	Maximum number of rules in a TR where, $NR(k_itemset) \leq k$
17	σ	Support value
18	c	Confidence value
19	$S_{i,j}$	Replica provider site where, $i = \{1, 2, \cdots , M\}$, and M is the maximum number of replicas. $j = \{1, 2, \cdots , N\}$, and N represents the maximum number of attributes for each replica
20	M	Maximum number of Replica provider sites
21	N	Maximum number of attributes of a site
22	NHF	Network History File
23	CTR	Correlated Transferring Rules
24	BRS	Best Replicas Set
25	W	Number of Replicas in BRS
26	TR	Transferring Rules
27	AR^3	Associated Replicas Rule Repository
28	LHF	Logical History File with (0,1) values
29	EST	Efficient Set Technique
30	$OWPS$	One Way Ping Service
31	$DGRIS$	Dynamic Grid Resource Information Service
32	TRT	Transferring Rule Technique
33	$CTRT$	Correlation Transferring Rule Technique
34	$FIST$	Frequent ItemSets Technique
35	RLS	Replica Location Service
36	NMS	Network Monitoring Service
37	$CBRS$	Correlated Best Replicas Set
38	RHF	Reputation History File

Table 4.2: Notation

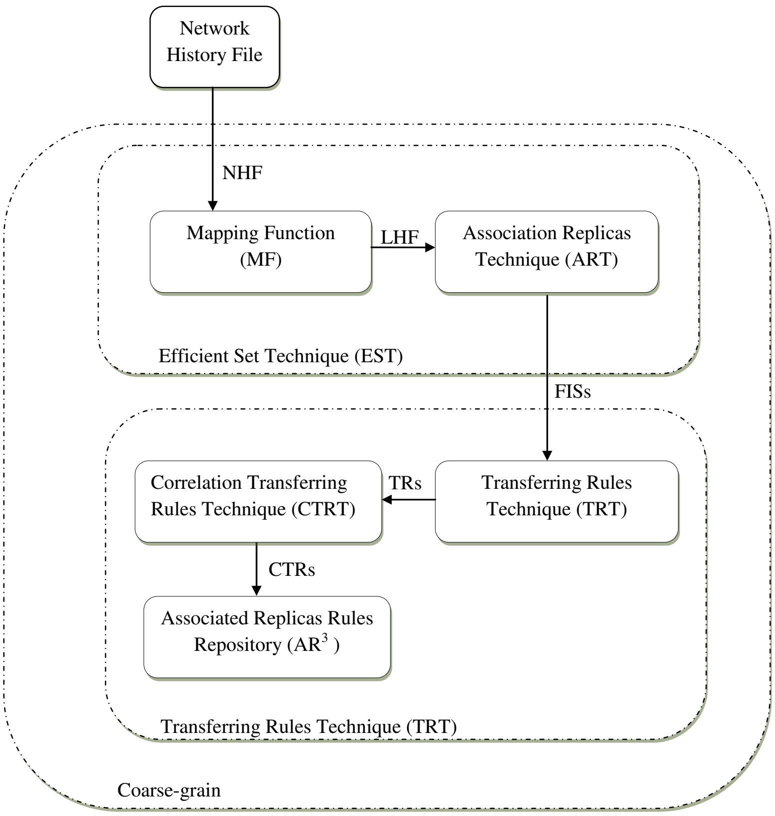

Figure 4.1: The Coarse-grain architecture

In brief, the Coarse-grain phase contains main four steps as shown in Figure 4.2,

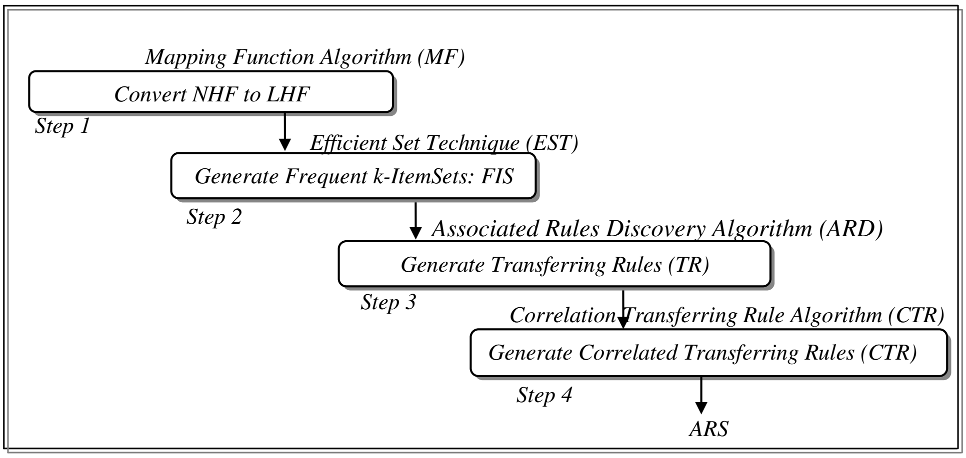

Figure 4.2: Four stages of Data Gird mining: Coarse-grain phase

The steps in the Figure 4.2 show the methods of extracting the replica sites with a good latency that work concurrently, and those that have a minimal cost (time) of getting the requested files. This is shown in the algorithm 4.1.

RTT	S_1	S_2	S_3	S_4	S_5	⋯	S_{29}	S_{30}
1	88	108	131	151	117	⋯	313	297
2	113	108	131	151	118	⋯	315	297
3	88.7	108	131	151	117	⋯	321	297
4	105	108	132	151	118	⋯	320	297
5	95.6	109	131	150	117	⋯	315	297
6	78.9	108	131	151	117	⋯	314	297
7	100	108	131	152	117	⋯	314	297
8	104	108	131	151	118	⋯	313	297
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
855	110	109	131	159	118	⋯	297	38.9
856	72.4	109	131	151	118	⋯	297	38.9
857	101	110	131	151	118	⋯	297	38.9
858	112	109	131	154	118	⋯	297	38.9
859	126	110	132	153	118	⋯	297	38.9
960	95.8	109	31	153	118	⋯	297	38.9

Table 4.3: Sample of Network History File (NHF) of 30 replicas of PingER report in 2nd Feb.2011

4.3.1 Efficient Set Technique (EST)

It is the first stage of the Coarse-grain phase. It is for finding all frequent itemsets. The algorithms which are used to do that are explained here:

4.3.1.1 Mapping Function (MF)

MF is used to convert Network History File (NHF) into Logical History File (LHF). LHF is a table that contains Boolean values (0,1). The mapping algorithm is shown in Algorithm 4.3 is applied on the round trip time (RTT) or Single Trip Time (STT) between computing site and replicas in an interval time $[t_0, t_z]$ as shown in Tables 4.3 and 4.4.

No.	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
1	0	1	0	1	0	1	0	1
2	0	1	0	1	0	1	0	1
3	1	1	0	0	0	1	0	1
4	1	1	0	0	0	1	0	1
5	0	1	1	1	0	1	0	1
6	0	1	0	1	0	1	0	1
7	0	1	1	1	0	1	0	1
8	0	1	1	1	0	1	0	1
9	0	1	1	1	0	1	0	1
10	0	1	1	1	1	1	0	1
11	0	1	1	1	1	1	0	1
12	0	1	1	1	1	1	0	1

Table 4.4: Sample of Logical History File (LHF)

4.3.1.2 Generate frequent item sets Technique (FIST)

In this section, one of the association techniques of Data Mining that called as Apriori algorithm [Puj01], is used to generate set of associated replicas. Apriori algorithm is used here to discover the hidden relationships of links connected replica providers such as, latency, for example. The result of applying Apriori algorithm is set of replica providers having a similar characteristic of links conditions (set of replica providers having uncongested links) at time of job execution.

The associated replicas are also called the frequent itemsets as shown in Table 4.5. The generated frequent itemsets are used to generate the association rules. Algorithm 4.4, shows the main steps of the generation frequent itemsets (FIS).

Algorithm 4.4 shows the main steps of the generation frequent itemsets (FIS). It uses NHF, a minimum support value $min-supp$, and the maximum number of replicas MR as parameters to generate transferring rules. It illustrates the main idea of the

No.	$k-Itemsets$	L_k = set of large k-item sets (set of items having minimum support)
1	$1-itemset$	$L_1 = \{\{S_3\},\{S_4\},\{S_6\},\{S_8\}\}$
2	$2-itemset$	$L_2 = \{\{S_3,S_4\},\{S_3,S_6\},\{S_3,S_8\},\cdots,\{S_6,S_8\}\}$
3	$3-itemset$	$L_3 = \{\{S_3,S_4,S_6\},\{S_3,S_4,S_8\},\cdots,\{S_4,S_6,S_8\}\}$
4	$4-itemset$	$L_4 = \{\{S_3,S_4,S_6,S_8\}\}$

Table 4.5: Example of Frequent Item Sets (FIS).

algorithm, in *line 1 & 2*, LHF is analyzed to generate the candidate and frequent $(1-itemset)$. This is done by calculating the support of each data item and comparing it to the minimum support. Loop from *line 4* to *line 20* in the algorithm is used to generate all frequent itemsets which are, $\{1,2,\cdots,K-itemset\}$ and save it in $ARS = \{\{L_1\},\{L_2\},\cdots,\{L_k\}\}$. Each iteration of the loop, say *iteration k*, generates the frequent *k-itemsets* based on the *(k-1)-itemsets* generated in the previous iteration. This loop continues until it is not possible to generate new itemsets or the number of items in an itemset exceeds the predefined maximum number of required replicas, MR. *Lines 5-12* generate all the new candidate frequent *k-itemsets* out of the frequent $(k-1)-itemsets$. *Lines 13-16* remove those candidate frequent $k-itemsets$ that do not fulfill the minimum support requirement. In *line 22* the algorithm returns all generated frequent-Item-Sets.

Table 4.5, shows the frequent itemsets generated from LHF, Table 4.4, by applying frequent itemsets generating algorithm with a minimum support equals $\{min-supp = 50\%\}$, and maximum number of associated replicas equals $MR = 4$.

4.3.2 Transferring Rules (TR)

This is the second stage of the Coarse-grain phase. It is used to generate a set of transferring rules for the AR^3 these rules will be used to select providers for identical Data Grid jobs (required files indexes are closed), generating TR and then evaluating. Transferring rules process is explained in the following subsections:

4.3.2.1 Generate Transferring Rules Technique (TRT)

The association rules are generated from frequent $k-itemsets$ after applying $DM-ARD$ and getting ARS. In our work these association rules are called *Transferring Rules (TR)*. Transferring rules are formed as $x \rightarrow y$, where x is a $(k-1itemset)$, y is a

$(1 - itemset)$ and $TR = x \cup y$.

Algorithm 4.5 shows the steps of generating the rules in TR algorithm. It takes a minimum confidence $(min - conf)$ as an input parameter and generates all possible rules from all itemsets. However, for each frequent itemset, the rules are generated as follows: one item of frequent itemset becomes the consequent of the rule, and all other items become the antecedent. Thus, a frequent $k - itemset$ can generate at most k rules. For example, suppose $\{S_1, S_2\} \rightarrow S_3, \{S_1, S_3\} \rightarrow S_2$, and $\{S_2, S_3\} \rightarrow S_1$.

After generating the rules, the rule confidences are calculated to determine if they are equal to or more than the minimum confidence. Only the rules with at least the minimum confidence are kept in the rule set called TR.
For instance, the confidence of rule $\{S_1, S_2, S_3\}$, can be calculated as follow:
$c = \sigma(\{S_1, S_2, S_3\})/\sigma(\{S_1, S_2\})$ If $c \geq min - conf$, then this rule is added to the rule set TR.

4.3.2.2 Generate Correlation Transferring Rules Technique (CTRT)

Typically the transactions differ in the number of present items (item value = '1'). Therefore, some transactions as shown in Table 4.4 might be necessary to be mined using one of Data Mining tools, to discover the hidden information that describe the relation among the replica's providers [Puj01].

Each transaction gives an information about co-occurring items in the transaction. Using this data one can create a co-occurrence table, which tells the number of times the objects (items) occurred together in the whole transactions. The co-occurrence table is useful to easily establish a simple rule such as the following example:
Rule 1= "Item 2 *(Antecedent) comes together with* Item 1 *(Consequent)in 10% of all transactions"*
In Rule 1, the 10% is a measure of the number of co-occurrences of these two items in the set of transactions, which is called a *support of a rule,* $\sigma(Rule1)$.

In other words, if the frequency of *Item 1* occurrence in the set of transactions is 10%, and that of *Item 2*, 20%, then the ratio of the number of transactions that support the Consequent the Rule 1 (10%) to the number of transactions that support the Antecedent part of the rule (20%) gives the confidence value of the rule (Rule 1). In this case the confidence of Rule 1 is:

$c(Rule1) = \frac{10}{20} = 0.5$

So, the confidence of the $Rule1$ is (0.5) and is equivalent to saying that when $Item1$ occurs in the transaction, there is a 50% chance that also $Item2$ will occur in the transaction. The most confident rules seem to be the best ones. But the problem arises, for example, if $Item2$ occurs more frequently in the transactions (let's say in 60% of transactions). In that case the rule might have lower confidence than the random guess! This suggests using another measure called improvement or lift ratio. That measure tells how much better a rule is at predicting the Consequent than just assuming the result. Improvement is given by formula [WHR98]

$$I(TR_k) = \frac{\sigma((Antecedent(TR_k)) \cup (Consequent(TR_k)))}{\sigma(Antecedent(TR_k)) \times \sigma(Consequent(TR_k))}$$

So, the improvement of Rule 1 is:

$$I(Rule) = \frac{0.1}{0.1 \times 0.2} = 5$$

When improvement is greater than 1 the rule is better than the random chance. When it is less than 1, it is worse. In our case Rule 1 is five times better than the random guess.

4.4 Performance evaluation of Coarse-grain

To evaluate the Coarse-grain phase, we take up a case study which is representing a typical Data Grid environment. Then, results of Coarse-grain phase are compared with other methods to evaluate the selection strategy used in Coarse-grain phase. In Coarse-grain phase, we use authentic data from a real Data Grid environment called the *LHC Computing Grid (LCG).*

4.4.1 The Case Study

In our simulation, the following scenario is considered, five files $\{f_1, f_2, f_3, f_4, f_5\}$ each 10GB in size are replicated and distributed on the 105 replica sites (each site holding a copy of a file is called as a replica site). We are assuming a replica selection broker to be used here, which is broadly like the one described in Chapter 3.

Consider the situation where a grid job J_1 is submitted to the broker which takes up the analysis of data stored in five files. To execute J_1 on the computing element S_0, in case that the requested files are not locally available in S_0, then the files must be transferred from replica providers to S_0. To know the replica providers of these files, replica location service is contacted by broker to get the IP addresses of providers. In our case study, the files are found in thirty distributed sites denoted by: $\{S_1, S_2, \cdots, S_{30}\}$, as can be seen in Figure 4.3. To select a set (group) of replica providers that can be asked to concurrently (at the same time) send the data to S_0 we apply the current coarse-grain technique.

Coarse-grain method uses PingER reports, network monitoring reports, to extract a Network History File (NHF) which is explained in the next section.

4.4.2 Analysis of PingER Monitoring Network Reports

The PingER report is used as a real network history file of the assumed replica providers and is collected by visiting the online distribution of historical data accessible at
"http://www-iepm.slac.stanford.edu/cgi-wrap/pingtable.pl" [slab]. The following script is used to get a specific monitoring data in specific date:

http://www-iepm.slac.stanford.edu/cgi-wrap/pingtable.pl?format=tsv&dataset =hep&file=packet$_l$oss&by=by-site&size=100&tick=hourly&year=2011&month=02 &day=22&from=CERN&to=WORLD&ex=none

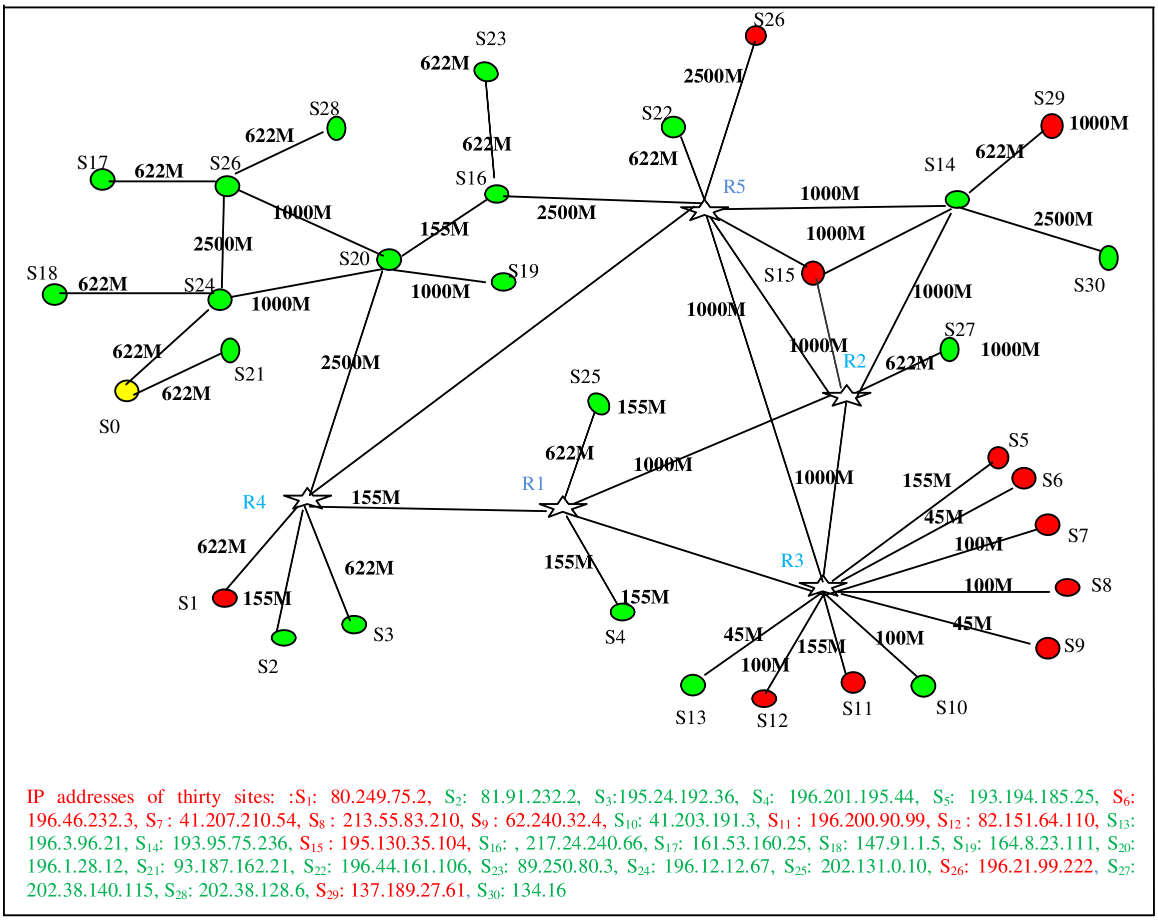

Figure 4.3: The EU Data Grid Test bed of 30 sites connected to CERN and the approximate network Bandwidth.

The example above will get you the hourly data from CERN to all the sites in the world for Feb 22, 2011. this can be imported into a text sheet and we use it in our work. The interval time can be changed to be for 30 minutes or less. The raw data format for the data produced by the monitor contains the Monitor Name (MN), Monitor Add (MA), Remote Name (RN) and Remote Add (RA) fields:

MN	MA	RN	RA	Time	Xmt	Rcv	Min	Avg	Max
xyz.edu	134.79.240.30	*abc.edu*	134.89.240.31	100	10	10	0.255	0.341	0.467

Table 4.6: PingER report fields

Table 4.7 shows the meaning of the notations used in the PingER report. The number of bytes in each ping packet can be 100 or 1000.

Xmt	is the number of ping packets sent
Rcv	is the number of ping packets received
Min	is the minimum response time for the packets sent (in milliseconds)
Avg	is the average response time for the packets sent
Max	is the maximum response time for the packets sent

Table 4.7: PingER report notation

Then for each ping response received the Sequence number is recorded, followed by the RTT for each ping for example:

01234567890.2870.3800.4670.3910.3270.3870.2910.3320.2550.299

After examining the PingER's report of the 30 replica providers $\{S_1, S_2, \cdots, S_{30}\}$, it has been noted that some sites at the same time are having stable network links as shown in Figure 4.6. In other words, at certain time of the day, some sites have Round Trip Times with almost constant values (good latency) as it is shown in Figure 4.4.

Figure 4.4 illustrates the status of the link between CERN site and site *waib.gouv.bj*, with IP (81.91.232.2) as it can be seen that the stability of the link varies from time to time. So the link between the two sites was stable at the beginning of *February 2nd 2011*, then it became unstable in the mid of the day, and after that it again became stable. The site *www.camnet.cm* with IP (195.24.192.36) has also a stable. link whereas the site *univ-sba.dz* with IP (80.249.75.2) has unstable link at the same time as shown in Figure 4.5. In this case, when the user sends a request to

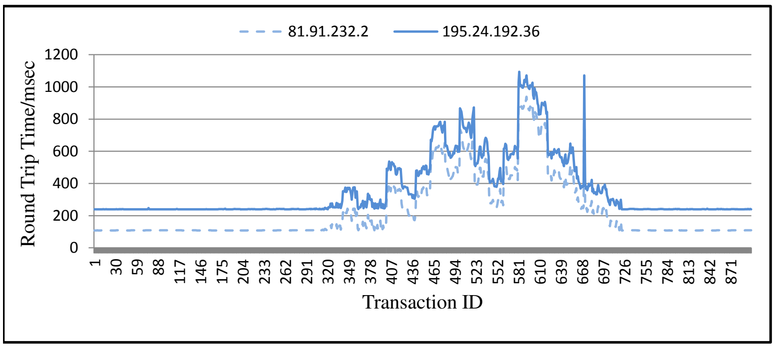

Figure 4.4: RTT between waib.gouv.bj and cern.ch

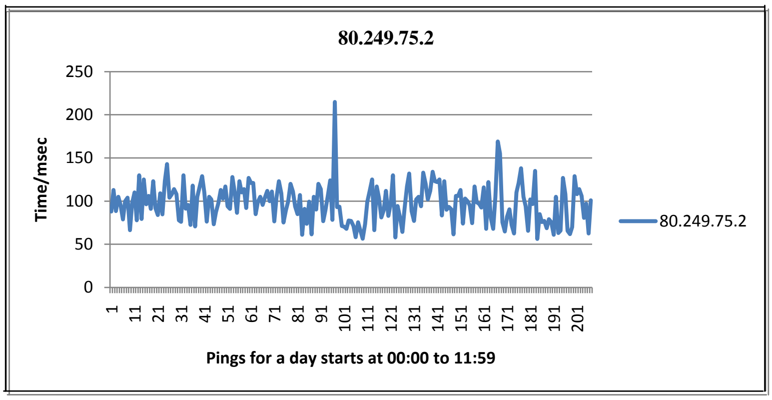

Figure 4.5: RTT of univ-sba.dz

our broker (2SOB) at the beginning of the day, both of the sites (waib.gouv.bj and www.camnet.cm) which had stabilized will be selected and appear in ARS after applying the Coarse-grain phase, whereas site univ-sba.dz will be discarded because it is unstable at that time as shown in Figure 4.7.

4.5 Implementation of the Coarse-grain phase

The implementation Algorithm 4.2 for our model is done by C++ program. The Coarse-grain phase implementation is divided into three parts, input part, processing part and output part which are explained in detail in the following subsections:

4.5.1 The input part

The input logs files are:

1. Network History File (NHF): In this file all round trips time between *cern.ch 192.91.244.6* and distributed sites for the date *(February 2nd 2011)* are saved.

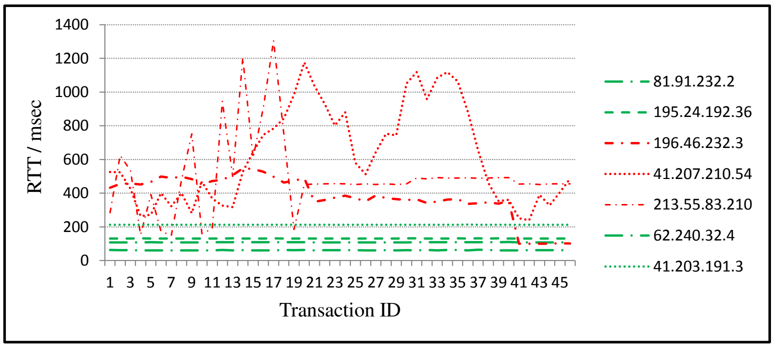

Figure 4.6: Latency history of seven data providers

2. The Bandwidth (BW), window size and other attributes can be taken RLS or NMS such as Iperf [ipe].

4.5.2 The processing part

The implementation of our model is done by writing a C++ program for Coarse-grain phase.

As we mentioned, association rules algorithm is the process of finding hidden relationships in data sets and summarizing these patterns in models [HT07]. The problem of mining association rules can be decomposed into two sub problems, first is finding ARS and second is about finding TR which are the output of Coarse-grain phase. The following functions are used in Coarse-grain algorithm to get ARS and TR.

1. Extracting RTT function: this function extracts only the RTTs values from PingER report and saves it in a text file called Network History File, NHF
2. Converting function: a mathematical standardization method is used to convert the real values of RTTs to logical values and save it in a text file called, Logical History File, LHF as shown in Algorithm 4.3
3. EST function: for discovering the Associated Replica Sites, ARS by executing the EST applying Algorithm 4.2
4. Rules Transferring function: for generating Transferring Rules applying Algorithm 4.5

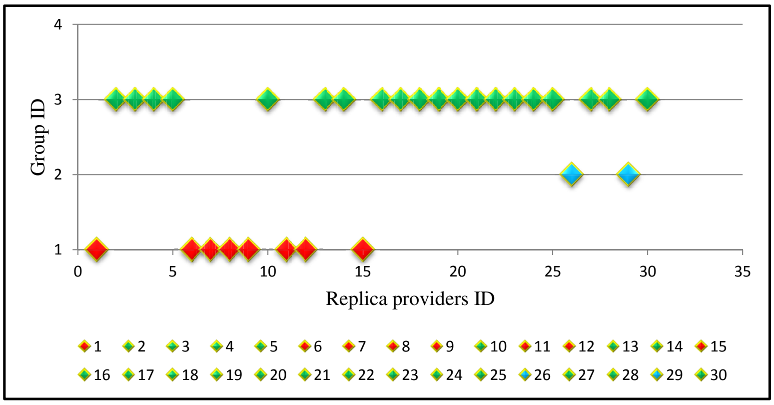

Figure 4.7: Output of Coarse-grain phase

5. Rules Testing function: this function is used to compute the improvement ratio (lift ratio) of the associated transferring rules using Algorithm 4.6

4.5.3 Result and Output file

In this section we test the output of the Coarse-grain with both stages as show in the following:

1. Generating Associated Replicas Set (ARS) using Efficient Set Technique (EST)

   In our example when, J_1 is submitted to our broker, the two phased service oriented broker at (02:00 A.M.), the output of Coarse-grain phase is shown in Figure 4.7. Coarse-grain phase separates the replica providers into three groups of providers, which are:

   Group1: Replica sites that have unstable links, lost packets percentage is high

   Group2: Replica sites that have stable links but high RTT (ARS)

   Group3: Replica sites with good latency, lost packets percentage is low

   The output of this stage is, ARS which is represented by Group 3 and contains the following set of provider sites:

   ARS=$\{S_2, S_3, S_4, S_5, S_{10}, S_{13}, S_{14}, S_{16}, S_{17}, S_{18}, S_{19}, S_{20}, S_{21}, S_{22}, S_{23}, S_{24}, S_{25}, S_{27}, S_{28}, S_{30}\}$

2. Generating Transferring Rules (TR): The frequent itemsets (FIS) are used to generate a set of associated transferring rules. The generated rules are used for

making decisions (to decide which providers work concurrently to send required files at a specific time). Table 4.8 shows some of transferring rules. The rules have a correlation value greater than one (CTRT= "yes"), will be kept in the Association Replicas Rules Repository (AR^3) to be used for similar jobs and files requests.

Rule#	*Conf%*	*Antecedent(a)*	*Consequent(c)*	$\sigma(a)$	$\sigma(b)$	$\sigma(a \cup b)$	*LiftRatio*	*CTRT*
1	100	S_3, S_5, S_8	S_4, S_6	46	97	46	1.97	*yes*
2	100	S_2, S_5, S_8	S_4, S_6	33	97	33	1.97	*yes*
3	100	S_2, S_3, S_5, S_8	S_4, S_6	30	97	30	1.97	*yes*
4	100	S_2, S_3, S_5, S_8	S_3, S_4, S_6	33	90	30	1.93	*yes*
5	90.91	S_1, S_4, S_5	S_2, S_3, S_7	33	92	30	1.89	*yes*
6	100	S_1, S_2, S_3, S_4	S_5, S_7	30	102	30	1.88	*yes*
7	66.39	S_2, S_5	S_4	119	139	79	0.916	*no*
8	85.94	S_2, S_6, S_8	S_3	64	181	55	9.116	*no*
9	71.9	S_3, S_5, S_6	S_2	57	154	41	0.89	*no*
10	56.52	S_2, S_3, S_5, S_7	S_4	92	139	52	0.78	*no*
11	50	S_1, S_7	S_4	66	139	33	0.69	*no*
12	51.56	S_2, S_6, S_8	S_5	64	144	33	0.687	*no*

Table 4.8: Sample of Transferring Rules Technique (TRT)

In the next section we compare the total time taken by file transfer process using different selections models.

4.6 Comparison between Efficient Set Technique (EST) with other methods while selecting a single site

In this section we compare the performance of the replica selection broker when it uses EST (our strategy) against when it uses other strategies while selecting a single provider site. The following notation are used in our simulation:

Table 4.9 shows the notation of all symbols being used in the algorithms.

No.	*Symbols*	Description
1	α	Total consumed time during consulting the replica catalog with the logical file names to get the physical file names
2	β	Total consumed time to probe the links between the computing site and replica sites to check availability of the file
3	γ	Total lookup time
4	$\rho(x)$	The run time of x strategy
5	$\tau(x)$	Total time of x selection method
6	$\delta(x)$	Total trace time of x selection method

Table 4.9: Notation used in the analysing the results.

In our simulation α denotes the time consumed during consulting the replica catalog with the logical file names to get the physical file names. The consumed time to probe the links between the computing site and replica sites is denoted by β. The probing time, β is used to check availability of the file and get recent information values of the condition links such as BW. The sum of both coefficients represents the lookup time which is denoted by γ, i.e.,

$$\gamma = \alpha + \beta$$

Replica broker has different results while applying different selection strategies. This section is to articulate the merits and demerits between the replica broker presented in this work and other works:

4.6.1 Comparison between (EST) and Traditional Model (TM)

In this section, comparison between total file transmit time when the replica selection broker uses EST strategy and TM selection strategies. In the traditional selection service model which is used in Globus [gloc]., replica provider site with the maximum bandwidth or the least number of Hops (routers) or the minimum Round Trip Time (RTT) is considered as the best replica provider. We observe that the total time is denoted by τ.

Total file transfer time using EST is,

$$\tau(EST) = \alpha + \beta + \rho(EST)\text{where, } \rho(EST) \text{ represents run time of EST.} \tag{4.1}$$

Further, the total file transfer time using TM is,

$$\tau(TM) = \alpha + \beta \tag{4.2}$$

Figure 4.8 shows the comparison between EST and TM where traditional method uses the least number of Hops and Figure 4.9 shows the comparison between EST and TM where traditional model uses highest Bandwidth. As it is observed in both figures, most of the times, the new technique adopted in 2SOB has a better performance. The reasons behind that, are: EST selects the stabilized links of replica sites whereas both of traditional models do not take into account links latency so, the selected site does not always be the best replica provider, as the link of the selected replica might be congested.

Analyzing the results: Figure 4.8 and Figure 4.9 show the following cases:

- TM performance is better than EST: Under rare circumstance, specially when with single replica provider is selected to send the file the total transfer time using TM is less than total transfer time using EST i.e., $\tau(TM) \leq \tau(EST)$. This happens when $\alpha(EST) \approx \alpha(TM)$ and $\beta(EST) \approx \beta(EST)$ and finally the link of the selected replica site using a TM be stable. The EST time is more than TM time by the time needed for executing EST i.e., $\rho(EST)$
- Equivalent: the total transmission time for both models (EST and TM) sometime be equal i.e., $\tau(EST) \approx \tau(TM)$. This case happens when time for both models (EST and TM) is equal, i.e., $\alpha(EST) \approx \alpha(TM)$ and also $\beta(EST) \approx \beta(EST)$ and sometimes, the link of the selected replica site using a TM be nearly stable
- EST performance is better than TM: This case happens most of the times. The main reason is that for EST, the selection decision always depends on the replicas having stable links. Stable links transmit files in less time since the number of lost packets is less in the stable links i.e., $\tau(TM) \geq \tau(EST)$

For clarity, let us assume this scenario in Figure 4.3, a job J_1 is submitted to the computing site, S_{27}. J_1, requires two files $\{f_1, f_2\}$ which are available in $\{S_3, S_5\}$. If the selection is done using the traditional Model which selects the replica with the least number of Hop count, then S_5 is selected as the best replica provider. S_5 sends the required file, because there is only two Hops, (R_3, R_2) between S_{27} and S_5, whereas there are three Hops, (R_1, R_2, R_4) between S_{27} and S_3. But,the required time to transfer the file from S_5 is more than the required time to move the file from S_3 because routers between S_{27} and S_3 are uncongested whereas the router R_3 between S_{27} and S_5 is congested. Same thing will happen when TM which uses the best site depending on the highest bandwidth between two sites. EST chooses S_3 to get the file because it has stable link with uncongested routers.

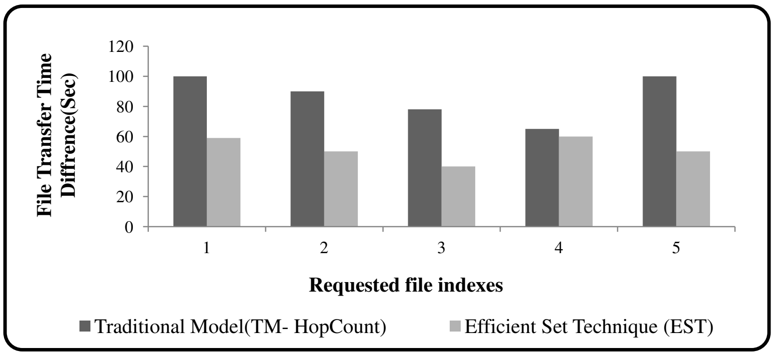

Figure 4.8: Comparison between EST and TM with HopCount criterion

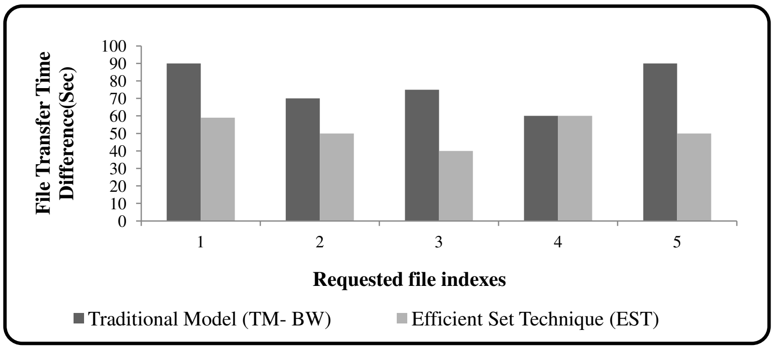

Figure 4.9: Comparison between EST and TM with Bandwidth criterion

4.6.2 Comparison between (EST) and Neural Network Model (NN)

If the replica selection broker uses a transfer time prediction model such as NN [RAB08] or regression models, then the transferring history file (this file contains all information about previous files transfers such as: the file size, the available network bandwidth, and transfer time), is needed to be a trace data, and used to train the prediction model.

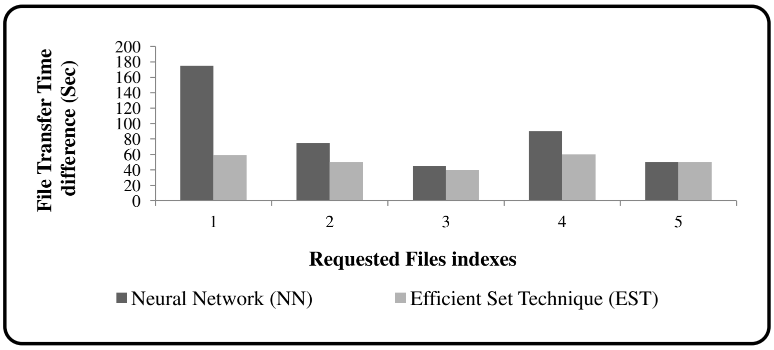

Figure 4.10: Comparison between EST and NN Model

Analyzing the results: As shown in Figure 4.10, both models have to consult the catalog and probe the links so, they both have a lookup time added to the total file transfer time, so:

$$\tau(NN) = \alpha + \beta + \rho, \text{ where, } \rho \text{ represents predicting time of the trained NN.} \tag{4.3}$$

As we see in Figure 4.10. The transfer time of requested files using neural network model takes more time than using EST because NN depends on two parameters which are: first, the size of requested file, and secondly, the bandwidth between the computing site and the replica site. Both of these parameters do not give a realistic picture of the dynamics of the network resources at the moment when the request for transfer was made. In Figure 4.10, The following cases can be seen:

- Equivalent: the total transmission time for both models (EST and NN) sometime is equal. The reason is, both models start with lookup time and sometimes the selected replica of the NN has a stable link
- Nonequivalent: Most of the times, the EST appears with more efficient than NN model. The main reason is that with EST the selection decision always depends on the replicas having stable links. Stable links transmit files in less time since the number of lost packets are less in the stable links. NN model does not take the latency of the network links into account

4.6.3 Comparison between (EST) and K-Nearest Neighbor Model (KNN)

In this section we illustrate the difference in the efficiency of the selection broker when using KNN and EST strategies. The transferring history file which contains information related to the previous jobs is needed by KNN strategy to start work.

4.6.4 Analysis of KNN

Since the KNN model needs at least 30 completed file transfers to begin, so the comparison is made up of two parts. First part related to time of the first 30 files transfer and the second related to the time of other files transfers starting from 31. As we mentioned, the first part of KNN uses TM and the comparison between TM and EST is done in above section 4.6.1. In Figure 4.11 and Figure 4.12, we compare the efficiency of the EST with right and wrong classifications of the KNN model of the second part, (starts from request number 31).

4.6.4.1 EST and KNN with the right classifications

Even though KNN model predicts a right classification, the efficiency of the selected site is less than EST because the Grid environment is dynamic where user requests and network latencies vary significantly, therefore the site selected by KNN may not be the best site for replica selection under this condition. In Figure 4.11, the following cases can be seen:

- KNN performance is close to EST: Under some circumstance, especially when with a single replica provider is selected to send the file the total transfer time using KNN is close to the total transfer time using EST.i.e., $\tau(KNN) \approx \tau(EST)$. This happens when $\gamma(KNN) \approx \gamma(EST)$ and the selected replica by a KNN has a stable link
- EST performance is better than KNN: This case happens most of the cases, EST appears to be more efficient than KNN. The main reason is that for EST the selection decision depends on the replicas which have stable links whereas the KNN does not care about the link latency. Stable links transmit files in less time since the number of lost packets are less in the stable links i.e., $\tau(EST) < \tau(KNN)$

In case of right selection the total time is,

1. $\tau(KNN - R) = \rho(KNN)$

4.6.4.2 EST and KNN with the Wrong classifications

Due to sudden change in network conditions, the KNN rule model may give wrong classifications. In [RAB08], where KNN is used as a selection replica strategy, the simulator of the KNN replica selection model switches to traditional model again if the KNN model gives five consecutive wrong classifications.

In Figures 4.11 and 4.12 we use $K = 5$ to depict the effect of right and wrong classifications on the total file transfer time for different types of file access patterns

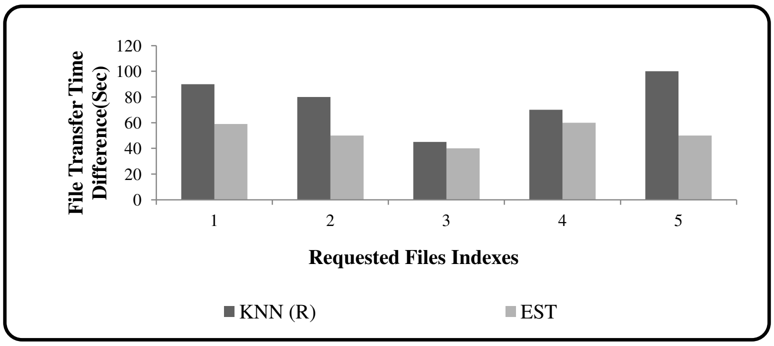

Figure 4.11: Comparison between EST and KNN with right selection

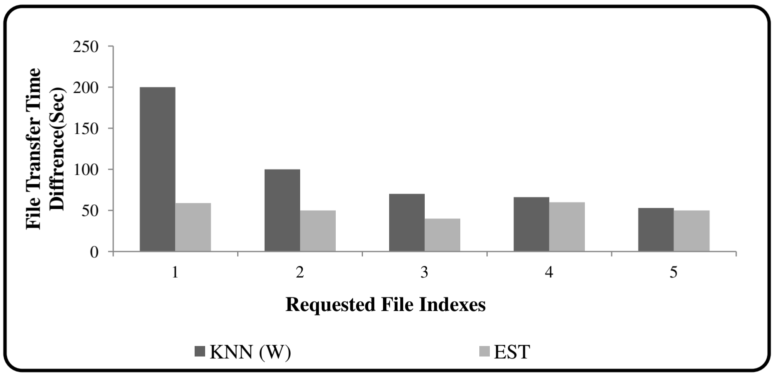

Figure 4.12: Comparison between EST and KNN with wrong selection

of the file requests such as sequential and random, for example.

As shown in Figure 4.12, the wrong classification makes the transfer time longer than the right one because, the time of wrong classification $\tau(KNN - W)$ comes from the summation of time spent to contact the predicted site which does not have the file any more and the replica lookup time of traditional model as shown in the following equation:

$$\tau(KNN - W) = \rho(KNN) + \tau(TM). \tag{4.4}$$

Figure 4.13 and Figure 4.14 depict the different transfer time among three models (Traditional, KNN and NN), as we see the best technique is EST because it takes the file from the site that has a copy of the file, one that is stable and has maximum number of data streams which carry the data file packets.

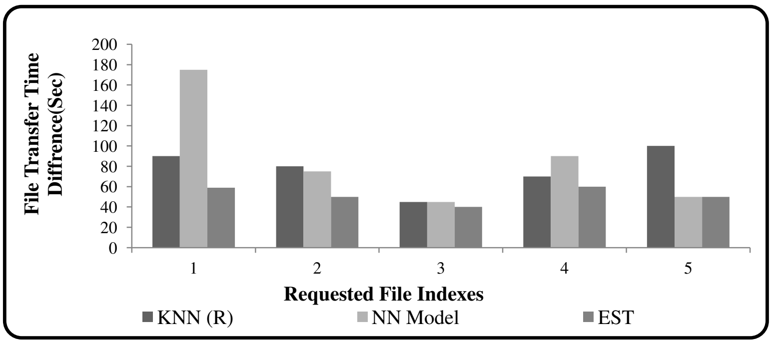

Figure 4.13: Comparison between EST, NN and KNN with right selection

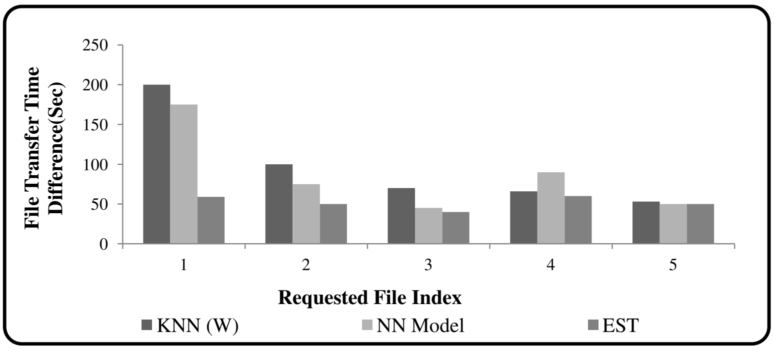

Figure 4.14: Comparison between EST and KNN with wrong selection

4.6.5 Comparison between (EST) and Genetic Algorithm (GA) model

In this section we illustrate the difference in the efficiency of the selection broker when using GA and EST strategies. A Rank based Genetic Algorithm is used as a new technique to enhance replica selection problem. The main objective of using GA is to reduce the time of searching appropriate replica provider who has a close matching to the user or application request. GA is used as a clustering method [AAHJ10]. The purpose of GA is grouping replica providers into k number of cluster centers $\{z_1, z_2, \cdots, z_k\}$. The clustering metric that has been adopted is the sum of the Euclidean distances of the points (replicas) from their respective cluster centers. Mathematically, the clustering metric CM for the k clusters $\{C_1, C_2, \cdots, C_k\}$ is given by:

$$CM = \sum_{l=1}^{k} \sum_{S_{i,j} \in C_l} ||S_{i,j} - z_l||$$

The task of GA is to search for the appropriate cluster centers such that the clustering metric CM is minimized. After clustering replica providers, Euclidean distance equation is used to find out all distances between users request and z_l, where, $\{l = 1, 2, \cdots, k\}$. Then cluster with minimum distance is selected since it contains the best provider.

4.6.6 Analyzing the results of GA

There are two slots of time in GA model. First the time of executing GA (time taken to trace GA with all possible generations), which is denoted by, $\delta(GA)$ and second slot of time is to run GA, which is denoted by, $\rho(GA)$. So, the total transfer time using GA is.

$$\tau(GA) = \gamma(GA) + \delta(GA) + \rho(GA) \tag{4.5}$$

GA has two options to work, which are:

- GA runs Online selection: in this case the GA strategy run when data grid job is submitted. In this case, the total transfer time using GA is,

$$\tau(GA - online) = \gamma(GA) + \delta(GA) + \rho(GA) \tag{4.6}$$

- GA runs Offline selection: in this case the GA strategy runs before data grid job is submitted.This assumption works properly when user/application does not specify a constraint time for execution the job. In this case, the total transfer time using GA is,

$$\tau(GA - offline) = \rho(GA) \tag{4.7}$$

The following subsections give the analysis of the results in detail.

4.6.6.1 EST and GA with the right selection

Even though GA model predicts a right selection, the efficiency of the selected site is less than EST because GA needs time to get results where in the Grid environment some of the replica providers attributes are dynamic such as (BW) and the network latencies vary significantly, therefore the site selected by GA might not always be the best site after some lapse of time. In Figure 4.11, the following cases can be seen:

- GA performance is close to EST: Under three special conditions, first, when the selection process is about only one provider and the second is, the work is done offline i.e., $\delta(GA) = 0$ and last, the selected provider by a GA has a stable link. Under these three conditions,

which is close to the total transfer time using EST.i.e., $\tau(GA) \approx \tau(EST)$

- EST performance is better than GA: This case happens most of the time, EST is more efficient than GA. The main reason is that EST chooses the replicas which have stable links whereas the GA does not care about the link latency. Stable links transmit files in less time since the number of lost packets are less in the stable links i.e., $\tau(EST) < \tau(GA)$

4.6.6.2 EST and GA with the Wrong selection

Due to sudden change in network conditions, the GA model may give wrong provider which may not have a copy of the file.

As shown in Figure 4.16, the wrong selection makes the transferring time longer than the right one because, the time of the wrong selection $\tau(GA-W)$ comes from the summation of time spent to contact the predicted site which does not have the file any more (the file has been deleted) and the replica lookup time of traditional model as shown in the following equation:
$\tau(GA - W) = \tau(GA) + \tau(TM)$.

There are two reasons that make GA unsuitable for work in Grid environment.

1. GA needs a long period of time to generate a sufficient number of generations and then to give a good selection result. This is not a suitable strategy for an online broker (jobs need to be immediately executed)

2. Dynamic nature of Grid make GA to fail since the optimizations needs to be started from scratch again for any change in the provider's attribute values

4.7 Comparison between EST and other methods while selecting a set of providers

To reduce file transfer times, more than one provider may be selected to concurrently send the required files. In this section we compare the performance of EST with other

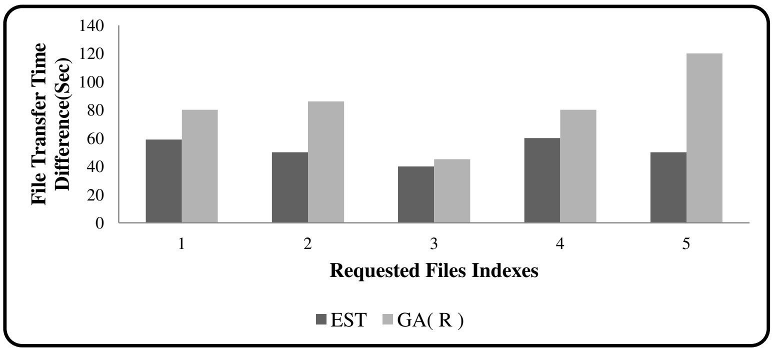

Figure 4.15: Comparison between EST and GA with right selection

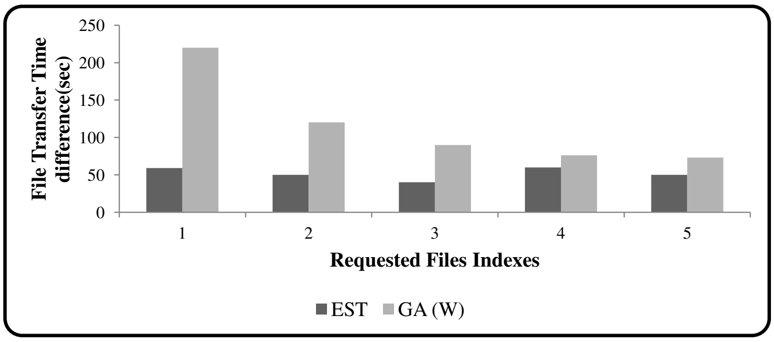

Figure 4.16: Comparison between EST and GA with wrong selection

replica selection strategies of data grid environment while selecting more than one provider (set of replicas). As it is noted in the Figure 4.17, EST gives the minimum file transfer time in most of the times. The reasons are:

1. Our technique, EST works well with two types of replica strategies: static and dynamic whereas most of the other models work only with static rather than dynamic replica strategy

2. Our technique selects a set of sites having similar characteristics at the point of file transfer, whereas others selects one site as a best replica's site. In case if these methods also apply the same concept of having more than one site, perhaps the same result will not be achieved

3. In previous methods, the selected replica site may or may not have the requested file since they depend on history of file requests which may be outdated information. Whereas in our method we do not have such a problem since we depend on current information of Local Replica Catalog

4. Some models like traditional method depends upon the Bandwidth alone or HopCounts alone which do not describe the real network condition, whereas we depend on the RTT or STT which reflects the real network conditions

5. EST is a scalable technique, because it can deal with the file transfer problems that arise suddenly. Let us take this example: in case of one of the selected providers becomes suddenly absence (leave the gird), using EST strategy, the broker can immediately equip with another provider (choose another provider from the associated replicas set (ARS)). Selects a new provider does not need to re-run the EST again. Compared with other methods where just a single replica provider is selected. However, to get another provider the other methods need to re-run again. In brief to get W providers using other mentioned methods, the selection strategy must be re-run W times. The re-run process adds an extra time to the total file transfer time which makes the method to be less efficiency than EST

4.8 Discussion and conclusion

In this chapter we presented a novel method known as Efficient Set Technique (EST) to be used as a selection strategy in the two phased Service Oriented Broker (2SOB). EST chooses the best set of replicas in the dynamic and static cases of data grid environment. We compared this method with the methods available in the literature for solving the same problem as shown in Figure 4.17. Followings are the observations:

1. Scalability: EST is a scalable technique. that is because while using EST a set of associated replica providers is determined. Where all of some of the associated providers cooperate to send the required file(s)

2. Availability: In most of the previous strategies, the selected providers might or might not have the required file(s) because they depend their selection on the previous file transfer requests. Using our approach there is no possibility of non-existence of the file(s) as the EST depends on the current list of processors with the help of the replica location service. will not have associated

3. Transfer speed: In all previous strategies when W providers are cooperated to

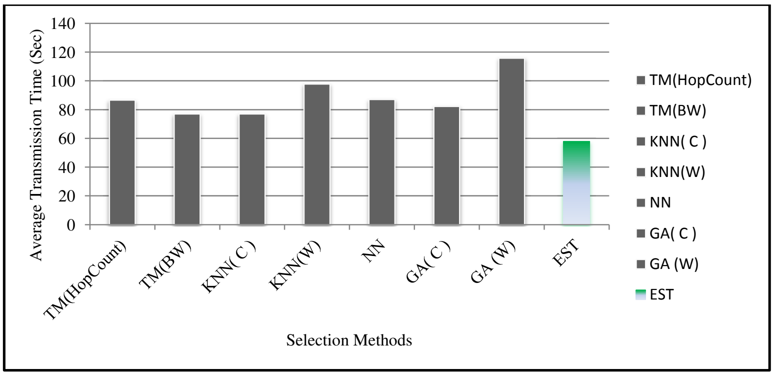

Figure 4.17: Comparison between EST and other selection strategies

concurrently send parts of file(s), all strategies need to re-run W times. The links of the W selected providers chosen (using previous strategies) will not have a relation (some of them are congested and some are not at the moment when the request for transfer was made. If the links are not stable the percentage of the packets being dropped is more and it needs retransmissions which adds extra time to the total file transfer time. In EST the W selected providers are associated and have stable links when the request for transfer was made

4. Easy to deploy: EST Strategy is an easy to deploy solution. As it is seen the EST strategy enables organizations to improve productivity and reduce transfer costs through easy-to-deploy solutions that work within existing Data Grid environments, delivering operational efficiencies. the output of EST is based on core services already available in the Data Grid such as, Network Monitoring Service and Replica Location Service for example

4.9 Summary

This chapter presents a brief overview of a Data Grid Mining model and some parallels of convergence between the Data Grid Mining and Data Mining, Association Rule Mining, ARM is explained with the Coarse-grain phase model. The Notation and the specification of the problem followed by the steps of Coarse-grain algorithm. The sub algorithms and functions of the coarse-grain model such as Data Mining-based Associated Replicas Discovery (DM-ARD) in detail are given. Finally, performance evaluation of Coarse-grain and implementation results comparing with other method are explained.

Algorithm 4.1 Main steps of the Coarse-grain phase: Data Grid Mining algorithm

Initialize:

[1] Specify a minimum confidence $min - conf$./* determined by the Broker depending on the data sets*/

[2] Specify a minimum support $min - supp$./* determined by the Broker depending on the data sets*/

Input:

[1] A Data Grid job (J_1)./*J contains name of required files*/

Output:

[1] Associated Replicas Set(ARS)

[2] Correlated Transferring Rules(CTR)

Begin

Step 1: Receive the description of Data Grid job (J)

Step 2: Converts user requirements to the QoS metrics, such as deadline of transmission time, physical file name, file size and required cost using Resource requirement interpreter

Step 3: An abstract resource requirement is passed to Dynamic Grid Resource Info. Services (DGRIS) unit to discover all the resources that are matching the user/application requirements with the dynamic attributes such as *Bandwidth*, *Delay* and *Cost*. For that, (DGRIS) contacts two core services, first RLS to get List of IP addresses of replica providers then, NMS to get network monitoring reports

Step 4: $ReplicaManagement$ unit has two processes, first is to compile the network monitoring reports to form NHF, /* NHF is a database table with:

Rows represented by either Single Trip Time STT values or Round trip time RTT values within an interval time $[t_0, t_z]$ whereas, /* depends on the NMS weather uses one way pings (STT) or round trip time pings (RTT) */

Columns represented by Identification of Replica Sites ($IDRS$), after that the NHF is sent to the Coarse-grain phase unit. The second process is to organize the list of replicas with their standardized attributes values denoted by: S_{ij}, to be sent to the Fine-grain phase unit

Step 5: Call $DM - ARD(in : NHF, min - conf, min - supp; out : ARS, CTR)$

Step 6: Call Fine-grain phase strategy (in: ARS, RHF, out: BRS) (will be explained in the next chapter)

Step 7: Select CBRS from BRS by checking CRT in AR^3

Step 8: Resource reservation interface is used to reserve the resources of CBRS

Step 9: Contact Transfer Service such as GridFTP to transfer required file(s) from multiple providers CBRS into computing site

Step 10: Release resources of the unselected replicas

Step 11: Get a new job

End

Algorithm 4.2 Data Mining-based Associated Replicas Discovery (DM-ARD) Algorithm

Input:
[1] NHF (see Table 4.3)
[3] $min - conf$
[4] $min - supp$
Output:
[1] Associated Replicas Set (ARS)
[2] Correlated Transferring Rules (CTR)
Begin
Step 1: Call NHF-Mapping(in:NHF, out:LHF) to convert $NHF \leftarrow LHF$./* convert real values of data base into Boolean values see Table 4.4 */
Step 2: Call EST(in:LHF,min-conf,min-supp; out:FIS) to get associated replicas set (ARS)
Step 3: Call TRT(in:ARS,min-conf; out:TR) to get set of transferring rules (TR)
Step 4: Call CTRT(in:TR ,min-conf; out:CTR) to get (CTR) and save it in AR^3
Step 5: Call Fine-grain to extract BRS
End

Algorithm 4.3 NHF-Mapping Algorithm

Initialize:
[1] l /*predefined window size*/
[2] j /*location of the sliding window*/
Input:
[1] NHF (see Table 4.3)
Output:
LHF
Begin
Step 1: Calculate the Mean of $STT_{i,j}$

$$MSTT_{i,j} = \frac{\sum_{k=j}^{l-1+j} (STT_{k,j})}{l}$$

Step 2: Calculate the Standard deviation of $STT_{i,j}$

$$STDEV_{i,j} = \sqrt{\frac{\sum_{K=j}^{l-(1+j)} (STT_{i,k} - MSTT_{i,j})^2}{l}}$$

Step 3: Calculate $Q_{i,j} = (\frac{STDEV_{i,j}}{MSTT_{i,j}}) * 100$

Step 4: Find a Coefficient Variation of Replica provider, j using: $AV_i = \frac{\sum_{j=i}^{M}(Q_{i,j})}{M}$, where M is the number of replicas,
AV_i represents average of the variation of $Q_{i,j}$
Step 5: Classify links into stable and unstable using the condition:
$IF(AV_i \leq Qi, j)$ then $LV = 0$ Otherwise $LV = 1$ /* LV represents a mapped Boolean Value of each STT*/
Step 6: LHF:=LV
End

Algorithm 4.4 Efficient Set Technique (EST): Algorithm to Generate Frequent Itemsets

Input:
[1] LHF
[2] $min - conf$
[3] $min - supp$
Output:
ARS
Begin
Initialize:
[1] $k = 1$
[2] C_1**= all the** $1 - itemsets$
Step 1: Read LHF **to count the support of** C_1 **to determine** L_1
C_1**= Candidate itemsets of order** $1 - itemsets$ **/*set of sets with single site*/**
L_1=$\{\forall(1 - itemsets) \in C_1 and \sigma(1 - itemset) \geq min - supp\}$
Step 2: $k = 2$ /* k represents the pass number */
Step 3: Read MR /*Maximum number of replicas determined by the broker*/
Step 4: While ($L_{k-1} \neq \emptyset$ and $k + 1 \leq MR$) do
Step 5: begin
Step 6: $C_k = \emptyset$ /* start generating candidate-itemsets */
Step 7: for all itemsets $l_i \in L_{k-1}$ do
Step 8: for all itemsets $l_j \in L_{k-1}$ do
Step 9: if $l_i[1] = l_j[1] \wedge l_i[2] = l_j[2] \wedge \cdots \wedge l_i[k-2] = l_j[k-2] \wedge l_i[k-1] < l_j[k-1]$,
then $c = l_i[1], l_i[2], \cdots l_i[k-1], l_j[k-1]$
Step 10: $C_k = C_k \cup \{c\}$
Step 11: for end
Step 12: for end /* end of generating candidate-itemsets */
Step 13: for all candidate itemsets $c \in C_k$ /* start pruning process */
Step 14: for all $/(k-1) - subsetsdofc/$ do
Step 15: if ($d \notin L_{k-1}$) then ($C_k = C_k - \{c\}$) /*Delete c from C_k*/
Step 16: for end /* end pruning process*/
Step 17: for all transactions $t \in T$ do
Step 18: L_K= All candidates in C_K with minimum support
Step 19: $k = k + 1$
Step 20: While end
Step 21: end ART
Step 22: Return $ARS = \cup_k L_k$
End

Algorithm 4.5 Transferring Rules Technique (TRT):
Algorithm to generate the association transfer rules (TR)

Initialize:
[1] $TR = \emptyset$
[2] $k = 2$
Input:
[1] $ARS = L_k$, List of Associated Replicas Set of order k
[2] $min - conf$
Output:
TR
Begin
Step 1: While ($L_k \neq \emptyset$) do
Step 2: for each itemset ($l_i \in L_k$) do
Step 3: for each itemsets $l_i[j] \in l_i$ do
Step 4: if ($\frac{s(l_i)}{s(l_i - l_i[j])} \geq min - conf$) then ($TR = TR \cup \{l_i - l_i[j] \rightarrow l_i[j]\}$)
Step 5: for end
Step 6: for end
Step 7: $k = k + 1$
Step 8: While end
Step 9: Return TR
End

Algorithm 4.6 Correlation Transferring Rule Technique (CTRT): Algorithm to compute the Improvement (lift) of the association transfer rules

Initialize:
[1] $CTR = \emptyset$
[2] $k = 1$
Input:
[1] TR
[2] NR
[3] LHF
Output:
CTR
Begin
Step 1: While ($k \neq NR$) do
Step 2: for each rule ($TR_k \in TR$) do
Step 3: Compute σ(Antecedent(TR_k)$\cup$ Consequent(TR_k)) in LHF
Step 4: Computeσ(Antecedent(TR_k)) in LHF
Step 5: Compute σ(Consequent(TR_k)) in LHF
Step 6: Measure rule's correlation for each rule TR_k using an Improvement equation:
$I(Rule) = \frac{\sigma((Antecedent(TR_k)) \cup (Consequent(TR_k)))}{\sigma(Antecedent(TR_k)) \times \sigma(Consequent(TR_k))}$
Step 7: If ($I(Rule) \geq 1$) then, $CRT = CRT \cup TR_k$, (this indicates positive correlation Otherwise, it is negative correlation)
Step 8: for end
Step 9: $k = k + 1$
Step 10: While end
Step 11: Return CTR
End

Chapter V

Fine-grain phase for Best Replicas Set Discovery

In this chapter we adapt a powerful matching technique using Grey based Rough Set Theory. The main objective of this chapter is to extract the set of best replica providers called *Best Replicas Set* (BRS) from the *Association Replicas Sites* (ARS) that was computed by Coarse-grain phase in the previous chapter, the BRS is a set of sites match the requirements of the user or application in data grids .

5.1 Introduction

As we mentioned in the previous chapter, each replica provider has a vector of attributes. In most cases, values attributes are uncertain and are not described by the single numerical value, but are described by linguistic values which make the selection problem more difficult. To deal with the uncertainty in the replica selection problem [ZZLL09] we propose a selection strategy that works with uncertainty values.

Uncertainty values means that each attribute has upper and lower limits and belongs to one category. For example, *"Security"* attribute with [9-10] grey value belongs to *"Very Good"* category as it is shown in Table 5.3.

Here we investigate this problem of finding a suitable match-up between the replica provider sites and the requirement of users. The M registered replica provider sites are denoted by: S_i, where, $i = \{1, 2, \cdots, M\}$, $M \geq 1$. Each replica (copy of requested file) has a vector of attributes determined by replica site. A a set of N properties used to describe the M registered replicas of the set S_i. $A = \{a_j, j = 1, 2, \cdots, N\}$, $N \geq 2$, where N represents the number of attributes for each replica.

Q is a set of N registered resource properties relevant to a user request R in terms of resource ontology the irrelevant properties have been removed, $Q = \{q_1, q_2, \cdots, q_K\}$,

$K \geq 1$, and $Q \subseteq P$.

R is a set of L requested replica properties with their weights, $R = \{(r1, w1), (r2, w2), \cdots, (r_L, w_L)\}$, $L \geq 1$. Therefore the replica selection problem can be transformed to a nearest match problem between R and S_i. There are well known approaches to solve the nearest match problem like artificial neural networks.

Actually, the best replica provider selection problem is also a *Multiple Attribute Decision Making* (MADM) problem [DBX05], [LJ05], and it is known that the Rough set theory (RST) [Sko85], [Sko85] is a widely used tool in data mining and knowledge discovery to solve a MADM problem.

Presently in literature, the rough set approach and also fuzzy-based approach have also been used to deal with the suppliers selection problem under uncertainty [ZZLL09]. In our work, we propose a strategy that uses a powerful tool called *Grey based Rough Set Theory (GRST)*. To the best of our knowledge, GRST is being used first time in replica selection problem of data grids.

The advantage of GRST system over others (rough set and fuzzy theory) is that, in the rough set theory work, the attribute values must be known precisely. Generally, decision makers' judgments on attribute often does not estimate an exact or fixed numerical value [BZ70]. In addition, the alternatives of ideal replica providers are decided by the lower approximation, so the rank of each ideal replica provider is equal. Therefore it is difficult to select the most ideal replica provider. The advantage of using grey theory over fuzzy theory is that grey system theory considers the condition of the fuzziness. That means, grey system theory can flexibly deal with the fuzziness situation. This chapter explains how a grey-based rough set approach is used in the Fine-grain phase.

5.2 Preliminary Concept

This section is to give a preliminary concepts of RST and GST theories that are used in the discovering strategy of Fine-grain phase.

5.2.1 Rough Set Theory (RST)

Rough set theory [Sko85] which is proposed by Pawlak in 1982, is an extension of conventional set theory that supports approximations in decision making. The rough set itself is the approximation of a vague concept set by a pair of precise concepts, called lower and upper approximations, which are a classification of the domain of interest into disjoint categories. The lower approximation is a description of the domain objects which are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset.

Definition 1: Information system framework [Sko85]

Let $I = (\mathbb{U}, \mathbb{A})$ be an information system (attribute-value system), where $\mathbb{U}$ is a non-empty set of finite objects (the universe) and $\mathbb{A}$ is a non-empty, finite set of attributes such that $a : \mathbb{U} \rightarrow V_a$ for every $a \in \mathbb{A}$. V_a is the set of values that attribute a may take. The information table assigns a value $a(x)$ from V_a to each attribute a and object x in the universe $\mathbb{U}$.

With any $R \subseteq \mathbb{A}$ there is an associated equivalence relation IND(R).

$$\text{IND}(R) = \left\{(x, y) \in \mathbb{U}^2 \mid \forall a \in R, a(x) = a(y)\right\} \tag{5.1}$$

The relation $IND(R)$ is called a R-indiscernibility relation. The partition of $\mathbb{U}$ is a family of all equivalence classes of IND(P) and is denoted by $\mathbb{U}/\text{IND}(R)(or\mathbb{U}/R)$.

If $(x, y) \in \text{IND}(R)$, then x and y are indiscernible (or indistinguishable) by attributes from R.

Definition 2: Definition of a rough set [Sko85]

Let $X \subseteq \mathbb{U}$ be a target set that we wish to represent using attribute subset R; that is, we are told that an arbitrary set of objects X comprises a single class, and we wish to express this class (i.e., this subset) using the equivalence classes induced by attribute subset R. In general, X is not expressed exactly, because the set may include and exclude objects which are indistinguishable on the basis of attributes R. However, the target set X can be approximated using only the information contained within R by constructing the $R - lower$ and $R - upper$ approximations of X:

$$\underline{R}X = \{x \mid [x]_R \subseteq X\} \tag{5.2}$$

$$\overline{R}X = \{x \mid [x]_R \cap X \neq \emptyset\} \tag{5.3}$$

Definition 3: Lower approximation and positive region [Sko85], the $R-lower$ approximation, or positive region, is the union of all equivalence classes in $[x]_R$ which are contained by (i.e., are subsets of) the target set. The lower approximation is the complete set of objects in $\mathbb{U}/R$ that can be positively (i.e., unambiguously) classified as belonging to target set X.

Definition 4: Upper approximation and negative region [Sko85], the $R-upper$ approximation is the union of all equivalence classes in $[x]_R$ which have non-empty intersection with the target set the union of the three equivalence classes in $[x]_R$ that have non-empty intersection with the target set. The upper approximation is the complete set of objects that in $\mathbb{U}/R$ that cannot be positively (i.e., unambiguously) classified as belonging to the complement ($\overline{X}$) of the target set X. In other words, the upper approximation is the complete set of objects that are possibly members of the target set X.

The set $\mathbb{U} - \overline{R}X$ therefore represents the negative region, containing the set of objects that can be definitely ruled out as members of the target set.

Definition 5: Boundary region [Sko85].

The boundary region, given by set difference $\overline{R}X - \underline{R}X$, consists of those objects that can neither be ruled in nor ruled out as members of the target set X. In summary, the lower approximation of a target set is a conservative approximation consisting of only those objects which can positively be identified as members of the set. (These objects have no indiscernible "clones" which are excluded by the target set.) The upper approximation is a liberal approximation which includes all objects that might be members of target set. (Some objects in the upper approximation may not be members of the target set.) From the perspective of $\mathbb{U}/R$, the lower approximation contains objects that are members of the target set with certainty (probability = 1), while the upper approximation contains objects that are members of the target set with non-zero probability (probability > 0), the R-borderline region of X is defined as:

$$BN_R(X) = \{\underline{R}\} \, and \{BN_R(X) \neq \emptyset\} \tag{5.4}$$

Definition 6: The rough set [Sko85].

The tuple $\langle \underline{R}X, \overline{R}X \rangle$ composed of the lower and upper approximation is called a rough set; thus, a rough set is composed of two crisp sets, one representing a lower boundary of the target set X, and the other representing an upper boundary of the target set X. The accuracy of the rough-set representation of the set X can be given [Sko85] by the following:

$\alpha_R(X) = \frac{|\underline{R}X|}{|\overline{R}X|}$

That is, the accuracy of the rough set representation of $X, \alpha_R(X), 0 \leq \alpha_R(X) \leq 1$, is the ratio of the number of objects which can positively be placed in X to the number of objects that can possibly be placed in X, this provides a measure of how closely the rough set is approximating the target set. Clearly, when the upper and lower approximations are equal (i.e., boundary region empty), then $\alpha_R(X) = 1$, and the approximation is perfect; at the other extreme, whenever the lower approximation is empty, the accuracy is zero (regardless of the size of the upper approximation).

Definition 7: Reduct and core [Sko85].

An interesting question is whether there are attributes in the information system (attribute-value table) which are more important to the knowledge represented in the equivalence class structure than other attributes. Often, we wonder whether there is a subset of attributes which can, by itself, fully characterize the knowledge in the database; such an attribute set is called a reduct.

Formally, a reduct is a subset of attributes $\text{RED} \subseteq R$ such that

- $[x]RED = [x]_R$, that is, the equivalence classes induced by the reduced attribute set RED are the same as the equivalence class structure induced by the full attribute set R.

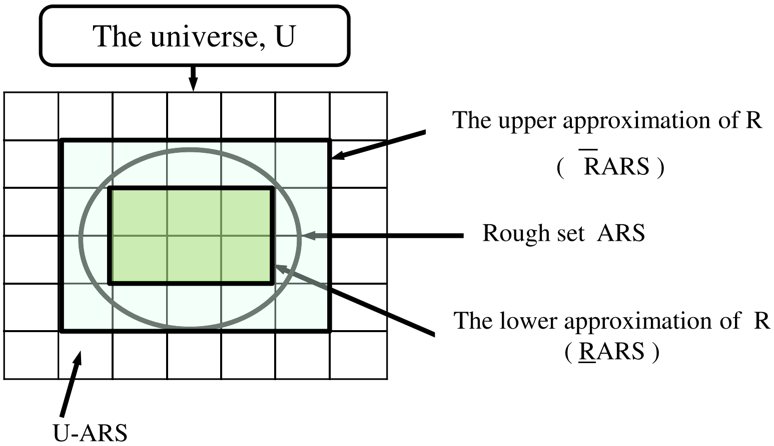

Figure 5.1: Lower and Upper Approximation

- the attribute set RED is *minimal*, in the sense that $[x]_{(\mathrm{RED}-\{a\})} \neq [x]_R$ for any attribute $a \in \mathrm{RED}$; in other words, no attribute can be removed from set RED without changing the equivalence classes $[x]_R$.

A reduct can be thought of as a *sufficient* set of features – sufficient, that is, to represent the category structure. The information system projected on just RED attributes possesses the same equivalence class structure as that expressed by the full attribute set. The reduct of an information system is not *unique* there may be many subsets of attributes which preserve the equivalence-class structure (i.e., the knowledge) expressed in the information system.

The set of attributes which is common to all reducts is called the *core*: the core is the set of attributes which is possessed by every legitimate reduct, and therefore consists of attributes which cannot be removed from the information system without causing collapse of the equivalence class structure. The core may be thought of as the set of necessary attributes, that is, for the category structure to be represented.

It is possible for the core to be empty, which means that there is no indispensable attribute: any single attribute in such an information system be deleted without altering the equivalence-class structure. In such cases, there is no essential or necessary attribute which is required for the class structure to be represented. The lower and upper approximations are shown in *Figure 5.1.*

5.2.2 Grey System Theory (GST)

Grey System Theory [YLN07], originally developed by Deng [DJL82], has become a very effective method of solving uncertainty problems under discrete data and incomplete information. In recent years, grey system theory has now been applied to various areas such as forecasting, system control, decision making and computer graphics [YLN06].

5.2.2.1 *Grey System, Grey Set and Grey Number Operation*

Definition 1: A grey system is defined as a system containing uncertain information presented by grey number and grey variables [YLN07].

Definition 2: Let X be the universal set. Then a grey set G of X is defined by its two mappings $\overline{\mu}_G(x)$ and $\underline{\mu}_G(x)$ [YLN07].

$$\{^{\overline{\mu}_G(x):x \to [0,1]}_{\underline{\mu}_G(x): \to [0,1]} \tag{5.5}$$

$$\otimes x = [\underline{x}, \infty) \tag{5.6}$$

Definition 3: If only the upper limit of x can be possibly estimated and x is defined as lower limit grey number.

$$\otimes x = (-\infty, \overline{x}] \tag{5.7}$$

Definition 4: If the lower and upper limits of x can be estimated and x is defined as interval grey number [YLN07].

$$\otimes x = [\underline{x}, \overline{x}] \tag{5.8}$$

Definition 5: The basic operation laws of grey numbers $\otimes x_1$=$[\underline{x}_1, \overline{x}_1]$ and $\otimes x_2 = [\underline{x}_2, \overline{x}_2]$ can be expressed as follows:

$$\otimes x_1 + \otimes x_2 = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2] \tag{5.9}$$

$$\otimes x_1 - \otimes x_2 = [\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2] \tag{5.10}$$

$$\otimes x_1 \times \otimes x_2 = [min(\underline{x}_1\underline{x}_2, \underline{x}_1\overline{x}_2, \overline{x}_1\underline{x}_2, \overline{x}_1\overline{x}_2), max(\underline{x}_1\underline{x}_2, \underline{x}_1\overline{x}_2, \overline{x}_1\underline{x}_2, \overline{x}_1\overline{x}_2)] \tag{5.11}$$

$$\otimes x_1 \div \otimes x_2 = [\underline{x}_1 \overline{x}_1] \times [\frac{1}{\underline{x}_2}, \frac{1}{\overline{x}_2}] \tag{5.12}$$

Definition 6: The Minkowski space distance of two grey numbers $\otimes x_1$ and $\otimes x_2$ is defined as

$$L(\otimes x_1, \otimes x_2) = [(\underline{x}_1 - \underline{x}_2)^p + (\overline{x}_1 - \overline{x}_2)^p]^{\frac{1}{p}} \tag{5.13}$$

In our study, $p = 2$ is used. It represents Euclidean grey space distance [YLN07].

The *Grey Relational Analysis (GRA)* is an important approach of grey system theory in the application of evaluating a set of alternatives in terms of decision criteria. In GRA, the data that contain same features are regarded as a sequence. As a tool of quantitative analysis, the GRA can be used to measure the relationship between two sequences by calculation their correlative degrees, which is called *Grey Relational Grade (GRG)*. The GRG is expressed by a scalar between 0 and 1. Up to now, the method has been used successfully in many fields. However, in conventional GRA, the data of sequences which are used as real numbers.We use a new GRA based on grey number to more flexibly analyze the uncertain relationship of system factors. *Definition 9. Considering a reference sequence [LYL+06, YLN06]* $\otimes x = \{\otimes x(1), \otimes x(2), \cdots, \otimes x(n)\}$ *and* m *comparative sequences* $\otimes x_i = \{\otimes x_i(1), \otimes x_i(2), \cdots, \otimes x_i(n)\}$, $i = \{1, 2, \cdots, m\}$, *where* $\otimes x_i(k)$ *represents the* Sk_th *attribute in* $\otimes x_i$,*where,* $\{k = 1, 2, \cdots, n\}$. *The* Grey Relational Coefficient *(GRC) of* $\otimes x_i$ *with respect to* $\otimes x_0$ *at the kth attribute is calculated as [YLN07]*

$$\gamma(\otimes x_0(k), \otimes x_i(k)) = \frac{\triangle_{max} - \triangle_{0i}(k)}{\triangle_{max} - \triangle_{min}} \tag{5.14}$$

Scale	Grey Values$\otimes v$	Normalized Values
1 Very Low (VL)	[0-1]	[0-0.1]
2 Low (L)	(1-3]	(0.1-0.3]
3 Medium Low (ML)	(3-4]	(0.3-0.4]
4 Fair (F)	(4-5]	(0.4-0.5]
5 Medium good (MG)	(5-6]	(0.5-0.6]
6 good (G)	(6-9]	(0.6-0.9]
7 Very good (MG)	(9-10]	(0.9-1]

Table 5.1: The scale of attribute ratings $\otimes v$ (1-10)

$$\triangle_{max} = \max_{\forall_i, \forall_k} \max L(\otimes x_0(k), \otimes x_i(k)) \tag{5.15}$$

$$\triangle_{max} = \min \min_{\forall_i, \forall_k} L(\otimes x_0(k), \otimes x_i(k)) \tag{5.16}$$

$$\triangle_{0i}(k) = L(\otimes x_0(k), \otimes x_i(k)) \tag{5.17}$$

$L(\otimes x_0(k),\otimes x_i(k))$ is the *Euclidean space distance* of $\otimes x_0(k)$ and $\otimes x_i(k)$ which is calculated by Equation *5.13.* The *GRG* between each comparative sequence$\otimes x_i$and the reference sequence $\otimes x_0$ can be derived from the average of *GRC*, which is denoted as

$$\varGamma_{0i} = \sum_{k=1}^{n} \frac{1}{n} \gamma(\otimes x_0(k), \otimes x_i(k)) \tag{5.18}$$

where $\varGamma_{0i}$ represents the degree of relation between each comparative sequence and the reference sequence. The higher degree of relation means that the comparative sequence is more similar to the reference sequence than comparative sequences [Wan07].

5.3 Fine-grain phase: Discovering BRS

A Fine-grain phase is a second phase of the proposed broker 2SOB. Fine-grain phase uses Grey-based Rough Set Theory (RST) concepts upon data sets [AWNR10d]. As shown in Figure 5.2, Fine-grain phase gets two input data files of ARS. The first data input file is, *Reputation History File* (RHF) and the second file contains list of attributes which are advertised by replica providers of Associated Replicas Sites (ARS) (set of sites that was selected by Coarse-grain phase).

RHF is a file that contains all previous information related to replica providers that the broker has dealt with. In another words, when the 2SOB selects a specific provider to get the required file, all information about this provider is stored in a tuple of RHF to be used later. Assigning reputation values is a job of decision maker(s)/system administrator(s) as shown in 5.2.

It means, whenever the process of monitoring file transfer finishes the decision makers evaluate the performance of the replica provider. Decision maker (DM)s always express their preferences on replica providers or on attributes of replica providers, which can be used to help ranking the replica providers or selecting the most desirable set. The preference information on replica providers and on attributes belongs to DMs' subjective judgments. Generally, DMs' judgment are often uncertain and cannot be estimated by the exact numerical value. Thus the selection problem of replica providers has many uncertainties and becomes more difficult.

Rough set theory [Sko85]- [Sko85] is a widely used tool in data mining and knowledge discovery. Up to present, the rough set approach has also been proposed to deal with the resource supplier selection problem [ZZLL09] . However, in the decision table of rough set theory, attribute values must be known precisely. In addition, the alternatives of ideal replica providers are decided by the lower approximation, so the rank of each ideal replica provider is equal. Therefore it is difficult to select the most ideal replica provider. The Lower/Upper approximations and reduct concepts are the processes of finding set of replica providers that match the requirements of the user or application in data sets. Approximations and Reducts are looking for patterns where one event is connected to another event.

As shown in Fig 5.2, the problem of discovering the Best Replicas Set (BRS) in Data Grids can be decomposed into three subproblems:

1. Classification: First task of the fine-grain phase is to classify ARS into an equivalence classes using lower approximation concept or positive region, $\underline{R}ARS$ (In our context, the positive region is called as the lower approximation) of Rough Set Theory [AWNR10d] (Equation 5.2). Positive region is the union of all equivalence classes in $[ARS]_R$ which are contained by (i.e., are subsets of) the target set, ARS. The $\underline{R}ARS$ is the complete set of replica providers in $\mathbb{U}/R$ that can be positively (i.e., unambiguously) classified as belonging to target set ARS

2. Extracting: Second task of fine-grain phase is extracting an ideal equivalence class $\underline{R}S^*$ from equivalence classes of $\underline{R}ARS$. In this task the Grey Relational Coefficient (GRC) (Equation 5.14) is used to to measure the correlative degrees between two sequences, values of user/application request VR and the positive region($\underline{R}ARS$). The class that has highest correlation (closest class to the requirements of user/application) is denoted by $\underline{R}S^*$

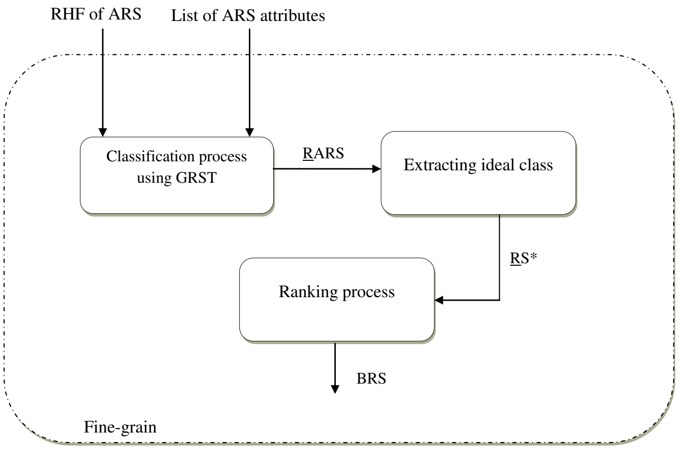

Figure 5.2: Fine-grain Architecture

3. Ranking and Filtering: Third task of fine-grain phase is ranking the providers of equivalence class $\underline{R}S^*$ using the Grey Relational Grade (GRG) as in Equation 5.13

All three stages represent of Fine-grain phase as shown in Figure 5.2. Table 5.2 lists the notations used to write an algorithm of Fine-grain phase.

Table 5.2 shows the notation of all symbols being used in the algorithms.

No.	*Symbols*	Description
1	$MADM$	Multiple Attribute Decision Making
2	DM	Decision maker
3	RHS	Reputation History File.
4	GRA	Grey Relational Analysis.
5	GRG	Grey Relational Grade.
6	RST	Rough Set Theory.
7	$GNIT$	Grey Normalized Information Table.
8	CT	Category Table.
9	GIT	Grey Information Table.
10	VR	Values of user/application request request.
11	$GRST$	Grey Rough Set Theory.
12	$K-GRST$	$K-means$ Grey Rough Set Theory.
13	R	User/application request.
14	GRC	Grey Relational Coefficient

Table 5.2: Notation

5.4 Proposed approach using Grey-based Rough Set Theory approach GRST

This section explains the Grey-based Rough Set Theory (GRST) approach of Fine-grain phase.

Assume that an grey information system of selection replicas sites is defined by $T = (U, A, V, f\otimes)$, where $U = \{S_1, S_2, \cdots, S_M\}$ represents the replica providers called the universe. $A = \{a_1, a_2, \cdots, a_N\}$ is a set of N attributes of replica providers. $f\otimes : U \times A \to V$ is grey description function. $T = (U, A \cup D, f\otimes)$ is called grey decision table, where D is a distinguished attribute called decision attribute. The elements of A are called condition attributes. The attribute ratings $\otimes v$ can be also expressed in grey numbers [LYL+06] by [1-10] scale shown in Table 5.1. The procedure of Fine-grain phase is summarized in Algorithm 5.1.

However, Fine-grain strategy can work with two options which are:

1. First when the Decision attribute(s) is available

2. Second when the Decision attribute(s) is absent

In Algorithm 5.1, we explain the first option when both Condition and Decision attributes are available [AWNR10d]. The Decision attributes (DA) are derived manually from the history file of replica provider sites RHS.

where $\otimes v_{ij}^K (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$ is the Kth attribute rating value and can be described by grey number $\otimes v_{ij}^K = [\underline{v}_{ij}^K, \overline{v}_{ij}^K]$

In the next section we explain the second option when the decision attribute is absent.

5.5 K-means GRST

This section is to extend the applicability of the GRST strategy to work when only the condition attributes are available. For such situations where the history of replica sites is unavailable which means the *Decision Attributes* are absent. Grey based Rough Set Theory is applied using replicas information only as an input data. In such a case, in our proposed strategy (GRST) we solved this by deriving the decision attribute using $Grey\ based\ K-means$ clustering algorithm upon the input data.

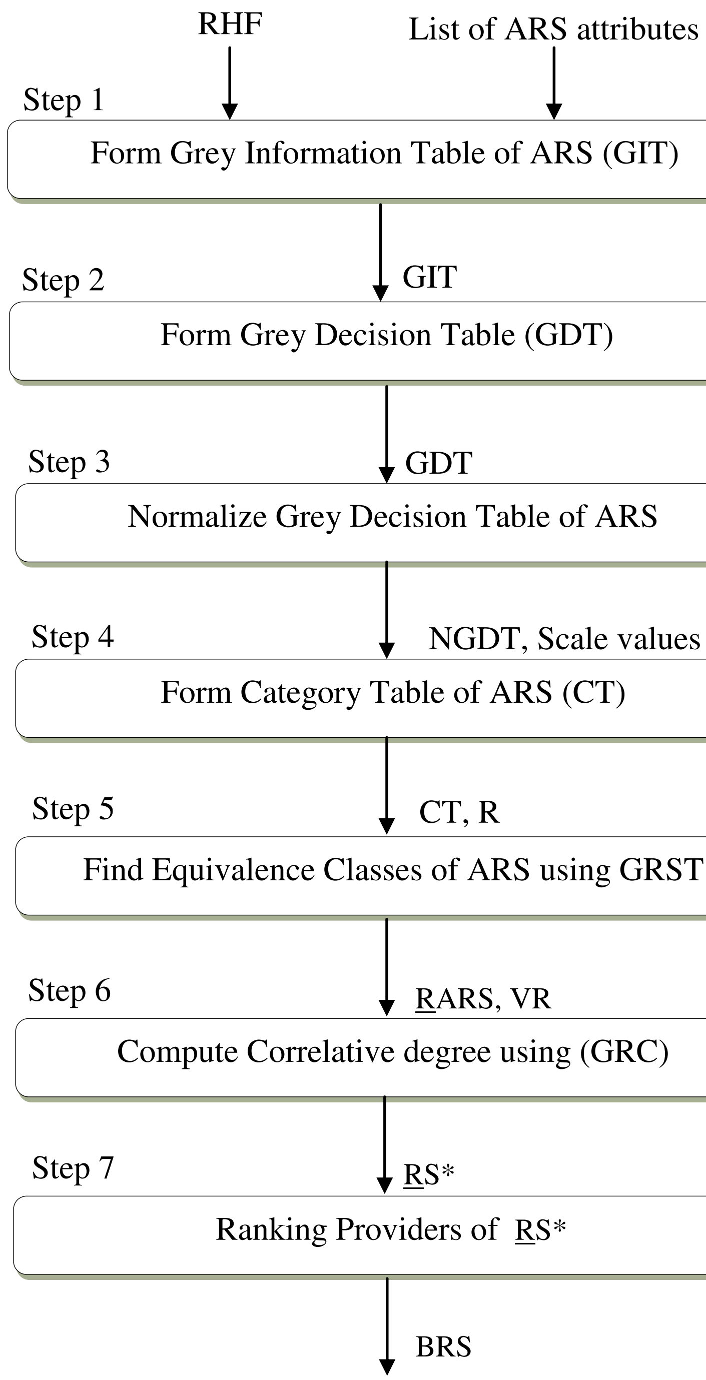

Figure 5.3: lower and upper approximations example

Grey based $K - means$ clustering algorithm is a $K - means$ algorithm but applied on Grey values. Each cluster label of (C_K) represents a class of Decision attribute in the *Decision Table* of the rough set. Here the replicas are clustered with respect to all replica attributes N. In the next subsection we update the algorithm of $K - means$ to be suitable with grey values.

5.5.1 Grey based K-means clustering algorithm

Here, we present an updated version of $K - means$ clustering algorithm that partitions the data sets into K clusters as shown in Algorithm 5.2. Each cluster comprises data-vectors with similar inherent characteristics. The overall outcome of this stage is the availability of $K - number$ of data clusters, which forms the basis for subsequent discovery of symbolic rules that define the structure of the discovered clusters.

5.6 *Application and Analysis*

In this section, we present two study cases. First is for GRST and second for K-GRST and the consideration is also described.

5.6.1 Case Study of GRST approach with a numerical example

In this case, the decision attribute(s) is available to run GRST for replica selection strategy.

Let us consider that there is a grey information system $T = (U, A, V, f\otimes)$ of replica providers selection. The grey decision table is expressed by $T = (U, A \cup D, f\otimes), U = \{S_i, i = 1, 2, \cdots, 10\}$ are as selected ten replica providers alternatives against six attributes $A = \{a_j, j = 1, 2, 3, 4, 5, 6\}$. The six attributes are: *availability quality, security quality, response time quality, price respectively, type of file and the frequency visited* as in Table 5.6 [ZZLL09]. The set of attributes that the selection decision depends on is referred by R i.e., where $R \subseteq A$. a_1, a_2 and a_3 are benefit attributes, the larger values are better, whereas a_3 and a_4 are cost attributes, the smaller values are better. As we can see in Table 5.4 the decision values $D_i, i = \{1, 2, \ldots, M\}$ are given by Replica Manger judgments. Replica manager judgment depends upon the history of the quality of services got from replica sites. Two values **"yes", "no"** are used for this attribute. Numerically **"yes", "no"** are represented as **"1", "0"**. The calculation procedures are shown as follows:

- **Step 1:** Get attribute rating values of the ten replica providers alternatives using resent entries of RHF. The most recent information of four entries are, $\{D_1, D_2, D_3, D_4\}$ has been formed by DMs to express the Quality of Service (QoS) of replica providers The results of attribute rating values are shown in Table 5.3

- **Step 2:** Establishment of grey decision table as in Table 5.4

- **Step 3:** Normalization: Normalize attribute values of grey information table to form Grey Decision Table (GDT) as it is shown in Table 5.8

- **Step 4:** In the GDT, some objects may have the same attribute values several times. The relation between these objects are called an indiscernibility relation for subset or all set of attributes. In our example we can see this relation between (S_4 and S_9) and also between (S_8,S_{10}) because they have same values of attributes $R = \{a_1, a_2, a_3, a_4\}$. This means, ($S_4$ and S_9) cannot be recognized by R attributes, so, to find elementary sets of ARS,$ARS = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}\}$ in space $R = \{a_1, a_2, a_3, a_4\}$, $R \subseteq A$, $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ the R-indiscernibility relation is used. The indiscernibility classes defined by ARS/R to describe the universe ARS with respect to R using Equation 5.1 :
  $\frac{ARS}{R} = \{\{S_1\}, \{S_2\}, \{S_3\}, \{S_4, S_9\}, \{S_5\}, \{S_6\}, \{S_7\}, \{S_8, S_{10}\}\}$

- **Step 5:** Select the ideal replicas set of sites S^* using lower approximation of rough set theory. Using lower approximation, Equation 5.2, set of replicas with the closest match to user's request is determined: $RS^* = \underline{R}X$, where $S^* = \{S_i | d_i = \backslash yes"\}$.This is to find all sites with Decision Attribute value is equal to "yes" i.e: $X_1 = \{S_i | D_i(S_i) = (1/yes)\} \rightarrow X_1 = \{S_1, S_4, S_5, S_7, S_9, S_{10}\}$.
  $\underline{R}X_1 = \{S_1\} \cup \{S_4, S_9\} \cup \{S_5\} \cup \{S_7\}$.
  $\overline{R}X_1 = \{S_1\} \cup \{S_4, S_9\} \cup \{S_5\} \cup \{S_7\} \cup \{S_8, S_{10}\}$.
  $X_2 = \{S_i | D_i(S_i) = (0/no)\} \rightarrow X_1 = \{S_2, S_3, S_6, S_8\}$.

$\underline{R}X_2 = \{S_2\} \cup \{S_3\} \cup \{S_6\}$.

$\overline{R}X_2 = \{S_2\} \cup \{S_3\} \cup \{S_6\} \cup \{S_8, S_{10}\}$.

The boundary relation can decide the roughness of the universe so that Equation 5.4 is used:

For X_1 the boundary set is: $BN_R(X_1) = \overline{R}X_1 - \underline{R}X_1 = \{S_8, S_{10}\}$ and for X_2 is $BN_R(X_2) = \overline{R}X_2 - \underline{R}X_2 = \{S_8, S_{10}\}$ as it is shown in Figure 5.1

- **Step 6:** Using Equation of ideal provider (explained in Step 8 of Algorithm 5.1, the most ideal referential replica provider S_0 against is:
  $S_0 = \{[0.93, 1.00], [0.88, 1.00], [1.00, 0.85], [1.00, 0.42]\}$

- **Step 7:** Using *Euclidean space distance*, Equation 5.13 calculates the distance between the ideal replica's provider and equivalence classes of positive regions of $\underline{R}ARS$. The first positive region when the Decision Attribute is **"yes"**, as shown in Table 5.10. The second positive region of $\underline{R}ARS$ when Decision Attribute is **"no"** as shown in Table 5.11

- **Step 8:** Calculation of GRC between comparative sequences $\underline{R}ARS$ with the ideal providers (S_0) using Equation 5.18, the values of GRG are shown in Table 5.12

- **Step 9:** An equivalence class with highest Γ is denoted by $\underline{R}S^*$

- **Step 10:** $\underline{R}S^*$ class has the most ideal replica's provider

- **Step 11:** Sort/Rank providers in $\underline{R}S^*$ class and select the one having the highest degree of GRC relation

  The higher degree of relation means that the comparative sequence is more similar to the (ideal provider)reference sequence than comparative sequences. In our example S_1 represents the best provider. In case more than one provider needed we sort the comparative sequences (set of providers in the lower approximation set) and then select as many providers as needed. Sorted list of providers in our example is : $\{S_1, S_8, S_{10}, S_7, S_9, S_4, S_5\}$

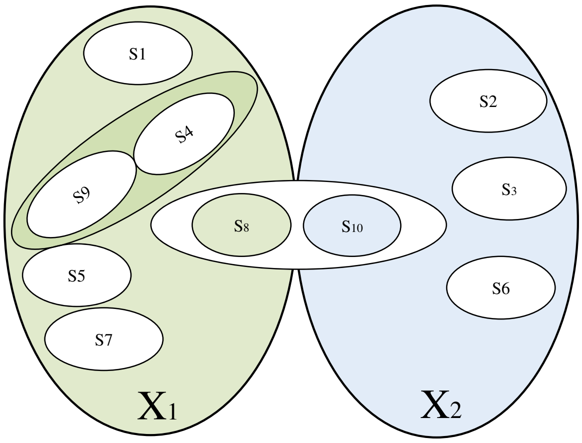

Figure 5.4: lower and upper approximations example

5.6.1.1 Simulations and Result of GRST

The RSES 2.2 (Rough Set Exploration System 2.2) software tool and (Matlab 7.6.0) are used for the simulation. They provide means for analysis of tabular data sets with use of various methods, in particular those based on Rough Set Theory [Sko85]. We simulate 99 replicas with different attributes and compare our work with the selection K-means-D-System proposed in [JSA09]. The authors of [JSA09] used K-means rule algorithm to select the best replica among multiple alternatives and the simulation. The results shown in Figure 5.5, implies that our proposed strategy covers the drawbacks mentioned by A. Jaradat and others in [JSA09], which risen up because of the non-consideration of the potential problems which are explained below.

Drawbacks: In K-means-D-system proposed by A. Jaradat in [JSA09] the clustering concept is weakly reflected by using number of clusters is equal to the number of replicas ($K = M$), whereas it should be ($K < M$). In [JSA09] the distance between the Ideal replicas, the one having best attribute values and other replica sites is a criterion of preference. Therefore knowing all distances, one can select the best replica site, which is the site with less distance value. But the system cannot accurately select the best replica in case of equal distances, which may occur in when the replicas have attribute values as in Table 5.14: Attributes/Site.

Using the Euclidean equation the distance will be the same and equal to (91.24144).

In this case it becomes impossible to select the best replica out of two equal ones. Our strategy covers this problem using grey numbers, so the same attributes can rarely occur even if they have the same linguistic degree. For example, if both replicas have *Good Security (G)*, we can observe when we look at Table 5.1 that the Good level taking many different values between [Sko85], [ZZLL09]. In case of finding the same security value they still may not have the same decision value, therefore the strategy can distinguish between them.

Second: The case of different values with the two equal distances. For example, in case there are two replicas (S_1, S_2), with the values of attributes in Table 5.15:

The distance using the Euclidean equation will be the same for both and equal to 70.7. Then it is hard again to select the best replica in this case. As we can see S_2 is far better than S_1 but the system cannot take a decision. If the system follows the authors of previous work [JSA09] selecting the replica arbitrary, it might select S_2, with low availability and security, at the same time with high cost and latency. This problem can be covered using normalization concept.There is another possible case of the system failure. Let us say again there are two replicas (S_1, S_2) with attribute values of Table 5.16

5.6.2 Case Study of K-GRST with a numerical example

In this section, we present a case study based on proposed approach to clarify the steps of our algorithm.
Case study: To select the best replica site by using our proposed algorithm use the following steps:

- Step 1: Call K-GRST(GIT, GDT) to get Grey Decision Table (GDT) as shown in Table 5.17. Each row belongs to one class of C_y
- Step 2: Compute Lower and Upper approximation sets of ARS for all cluster labels C_y, where $C_y = D_y$=1, 2./* In our study case K=2 */
  a. Get R= $\{A_1, A_2, A_3, A_4\}$ and ARS= $\{S_1, S_2, \cdots, S_{10}\}$
  b. Find ARS/UR= $\{\{S_1\}, \{S_2\}, \{S_3\}, \{S_4, S_9\}, \{S_5\}, \{S_6\}, \{S_7\}, \{S_8, S_{10}\}\}$
  c. Find $\underline{R}ARS$ and $\overline{R}ARS$ of C_y= $\{C_1, C_2\}$

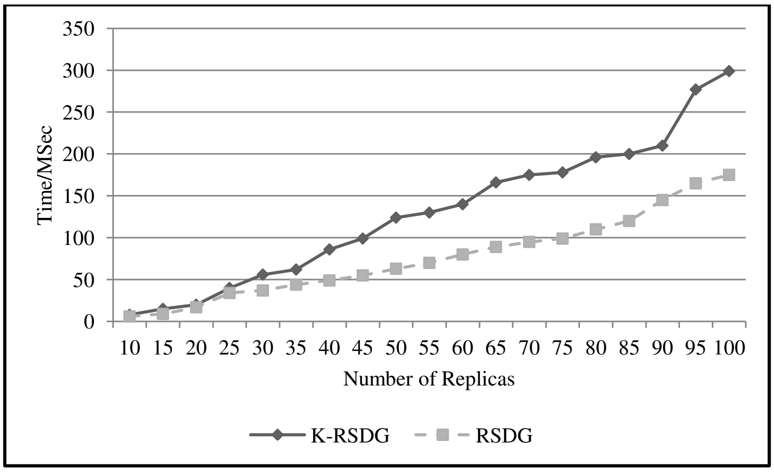

Figure 5.5: Comparison between Total execution time of $K - RSDG$ and RSDG

Now we compute the approximations for two cluster $\{C_1, C_2\}$. For the first cluster the $D = 1$ and,

X_1= $\{ARS_i|D_y = 1\} \rightarrow X_1 = \{S_1, S_4, S_5, S_7, S_9, S_{10}\}$ so, the lower approximation of X_1 is:

$\underline{R}X_1$= $\{S_1\} \cup \{S_4, S_9\} \cup \{S_5\} \cup \{S_7\}$, and the upper approximation of X_1 is:

$\overline{R}X_1$= $\{S_1\} \cup \{S_4, S_9\} \cup \{S_5\} \cup \{S_7\} \cup \{S_8, S_{10}\}$, and the boundary of X_1 is:

$BN_R(X_1) = \{S_8, S_{10}\}$

For the second cluster C_2 where $D = 2$ and,

X_2= $\{ARS_i|D_y = 2\} \rightarrow X_2$= $\{S_2, S_3, S_6, S_8\}$ so, the lower approximation of X_2 is:

$\underline{R}X_2$= $\{S_2\} \cup \{S_3\} \cup \{S_6\}$ and the upper approximation of X_2 is:

$\overline{R}X_2$= $\{S_2\} \cup \{S_3\} \cup \{S_6\} \cup \{S_8, S_{10}\}$, and the boundary of X_2 is:

$BN_R(X_2)$= $\{S_8, S_{10}\}$

5.6.2.1 Simulation and results $K - GRST$

In our simulation we used 99 replicas each replica has different attribute values. We compared our work with other previous strategies, such as GRST when the Decision attribute is available [AWNR10b]. The simulation results showed that the time to find the best replicas set in, $K - GRST$ is longer than the time of $GRST$ by the time of getting the Decision attribute (clustering time) as shown in Figure 5.5.

5.7 Conclusions

In the previous chapter, using Coarse-grain phase strategy an Associated Replicas Set (ARS) was generated. To find out a set of Best Replicas Set (BRS) from ARS, a novel selection strategy is used in the Fine-grain phase. Henceforth, we use the term "best replica" to express the highest level of QoS for both the user or application replica's requirements and the site features which stores these replicas. The criteria of the attributes are heterogeneous and conflicting with each other, making the problem quite complex to solve. Therefore the Grey-Rough Set Selection Theory (GRST) is used to select the best set of replica providers (BRS) with the closest match to the numeric and non-numeric user or application QoS requirements. Our proposed strategy GRST is suitable for solving the group decision-making problem under uncertainty environment in case the decision attributes are available or not.

Lower approximation concept being a classification of the domain, it works to distinguish between a group of replicas ($\underline{R}ARS$) with closest match to the user requirements. Then the selected providers BRS are requested to send the different parts of the file(s) to accelerate the execution of the job. The experimental results show that the proposed approach is reliable, reasonable and can help to select a replica in a more accurate way. The experiments are carried out on random data sets. Our results show better performance compared to others reported works.

5.8 Chapter Summary

In this chapter we have explained the Fine-grain phase of replica selection which represents the second phase of our proposed broker (2SOB). First we describe essential concepts from rough set theory. and grey system theory which include some basic definitions. We introduce the *Fine-grain* approach by grey-based rough set theory in both cases presence and absence of the decision attribute. After that, the application and analysis of proposed approach is introduced by an example of replica providers selection. Finally, conclusions are derived. In the next chapter we explain how the selection strategy of Fine-grain phase is upgraded to work in the storage management of resource broker of Cloud infrastructure.

Algorithm 5.1 Coarse-grain Fine-grain phase to discover BRS

Initialize:

Specify W /* determine number of replica providers when the user request is represented by the most ideal attributes of the providers */

Input:

[1] Associated Replica Sites (ARS) [2] User request with AR=$\{a_1, a_2, \cdots, a_Q\}$, $VR = \{vr_1, vr_2, \cdots, vr_Q\}$

Output:

[1] Best Replicas Set (BRS)

Begin

Step 1: Contact *Decision Maker (DM)* component to get $Reputation\ History\ File\ RHF$ of ARS providers. /*a committee of decision-makers identify some of attribute values of replica providers. using linguistic variables based on grey numbers It is shown in Table 5.3 */

Step 2: Contact $Replica\ Management$ system to get attribute values of ARS./* These attributes are given by replica providers*/

Step 3: Convert linguistic variables into grey values using the scaled values show in Table 5.1

Step 4: Establishment of $Grey\ Information\ Table$ (GIT) as shown in Table 5.4

Step 5: Establishment of *Grey Decision Table (GDT)* by set the Decision Attributes (DA): if(DA is available in GIT) then go to Step 6 /* The decision values $D_i, i = \{1, 2, \ldots, m\}$ from HRF. The final values are decided by Reputation Manager RM judgment. The real numbers of "yes", "no" are given as "1", "0" by the important degree of replica providers*/ else if (DA is not available in GIT) then call K-GRST(in:GIT, out: GDT)/*it is a procedure to derive DA*/

Step 6: Normalization of GDT. For benefit attributes use the following equations:
$\otimes v^*_{ij} = [\frac{\underline{v}_{ij}}{v^{max}_j}, \frac{\overline{v}_{ij}}{v^{max}_j}]$ $v^{max}_j = max_{1 \leq i \leq m} \overline{v}_{ij}$
For cost attribute, $\otimes v^*_{ij}$ is expressed as:
$\otimes v^*_{ij} = [\frac{v^{min}_j}{\overline{v}_{ij}}, \frac{v^{min}_j}{\underline{v}_{ij}}]$ $v^{min}_j = min_{1 \leq i \leq m} \underline{v}_{ij}$
/*The normalization method mentioned above is to preserve the property that the ranges of normalized grey number belong to [0, 1]. It is shown in Table 5.5*/

Step 7: Selection of ideal replica providers by grey-based rough set lower approximation [YLN06]. The lower approximation of ideal replica providers S_* are calculated as: $\underline{R}S_* = S_i \in U|[S_i]_R \subseteq S_*$, where $S_* = ARS_i|D_i = \backslash yes$"

Step 8: Get vector of user/application QoS attributes, $VR^* = \{vr^*_1, vr^*_2, \cdots, vr^*_Q\}$ or GRA is used to select the most ideal replica provider (S_0) against $\underline{R}S_*$ by: $S_0 = \{[max\forall_i \underline{v}^*_{i1}, max\forall_i \overline{v}^*_{i1}], [max\forall_i \underline{v}^*_{i2}, max\forall_i \overline{v}^*_{i2}], \cdots, [max\forall_i \underline{v}^*_{im}, max\forall_i \overline{v}^*_{im}]\}$ /* In case user determined QoS requirement of replica provider, vector of requirements is used as an ideal provider otherwise the vector of ideal replica provider is manually derived*/

Step 9: Calculation the relationship (Γ) between ideal provider, S_0 and providers in $\underline{R}S_*$ of GRG between comparative sequences $\underline{R}S_*$ with reference sequence S_0 using Equation 5.18 to get Γ

Step 10: Rank replica providers sites in Γ_R, where $p = \{1, 2, \cdots, |\underline{R}S_*|\}$

Step 11: Select highest W providers of ranked providers

Step 12: Contact GridFTP to get required files from selected providers W

End

a_j	S_i	D_1	D_2	D_3	D_4	$\otimes v_{ij}$
a_1						
	S_1	VG	VG	VG	VG	$[9.25, 9.900]$
	S_2	MG	MG	MG	MG	$[5.00, 6.00]$
	S_3	F	F	F	F	$[4.75, 5.00]$
	S_4	MG	MG	MG	MG	$[5.50, 5.50]$
	S_5	F	F	F	F	$[4.50, 5.00]$
	S_6	MG	MG	MG	MG	$[5.25, 6.00]$
	S_7	G	G	G	G	$[8.70, 9.00]$
	S_8	G	G	G	G	$[8.50, 9.00]$
	S_9	MG	MG	MG	G	$[5.50, 5.50]$
	S_{10}	G	G	G	G	$[8.50, 9.00]$
a_2						
	S_1	G	G	G	G	$[7.50, 8.50]$
	S_2	MG	MG	MG	MG	$[5.50, 6.00]$
	S_3	MP	MP	MP	MP	$[3.25, 4.00]$
	S_4	MP	MP	MP	MP	$[3.00, 3.50]$
	S_5	P	P	P	P	$[2.50, 3.00]$
	S_6	MP	MP	MP	MP	$[3.00, 3.50]$
	S_7	G	G	G	G	$[6.50, 7.50]$
	S_8	P	P	P	P	$[2.30, 3.00]$
	S_9	MP	MP	MP	MP	$[3.00, 3.50]$
	S_{10}	P	P	P	P	$[2.30, 3.00]$
a_3						
	S_1	G	G	G	G	$[6.50, 7.50]$
	S_2	F	F	F	F	$[4.75, 5.00]$
	S_3	MG	MG	MG	MG	$[5.25, 6.00]$
	S_4	G	G	G	G	$[6.50, 7.50]$
	S_5	G	G	G	G	$[6.50, 8.50]$
	S_6	F	F	F	F	$[4.25, 5.00]$
	S_7	G	G	G	G	$[7.50, 7.50]$
	S_8	G	G	G	G	$[6.5, 8.50]$
	S_9	G	G	G	G	$[7.75, 8.25]$
	S_{10}	G	G	G	G	$[1.25, 3.0]$
a_4						
	S_1	G	G	G	G	$[7.50, 8.00]$
	S_2	G	G	G	G	$[6.75, 8.25]$
	S_3	G	G	G	G	$[7.50, 9.00]$
	S_4	G	G	G	G	$[7.75, 8.25]$
	S_5	MG	MG	MG	MG	$[5.25, 6.00]$
	S_6	G	G	G	G	$[7.50, 9.00]$
	S_7	G	G	G	G	$[7.50, 8.00]$
	S_8	P	P	P	P	$[1.25, 3.0]$
	S_9	G	G	G	G	$[7.75, 8.25]$
	S_{10}	P	P	P	P	$[1.25, 3.0]$

Table 5.3: Attribute rating values for replica providers

Replicas providers	Condition attributes				Decision D
	a_1	a_2	$\dots$	a_n	
S_1	$\otimes v_{11}$	$\otimes v_{12}$	$\dots$	$\otimes v_{1n}$	d_1
S_2	$\otimes v_{21}$	$\otimes v_{22}$	$\dots$	$\otimes v_{2n}$	d_2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
S_m	$\otimes v_{m1}$	$\otimes v_{m2}$	$\dots$	$\otimes v_{mn}$	d_m

Table 5.4: The grey decision table

Replicas providers	Condition attributes				Decision D
	a_1	a_2	$\dots$	a_n	
S_1	$\otimes v^*_{11}$	$\otimes v^*_{12}$	$\dots$	$\otimes v^*_{1n}$	d_1
S_2	$\otimes v^*_{21}$	$\otimes v^*_{22}$	$\dots$	$\otimes v^*_{2n}$	d_2
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
S_m	$\otimes v^*_{m1}$	$\otimes v^*_{m2}$	$\dots$	$\otimes v^*_{mn}$	d_m

Table 5.5: The normalized grey decision table

Algorithm 5.2 Main steps of Grey $K-means$ algorithm

Initialize:
[1] $C_p = \emptyset$
Input:
[1] GIT /*Grey Information Table*/
[2] K /* Number of clusters*/
Output:
[1] Grey Decision table GDT
Begin
Step 1: Randomly select K replicas of S_i to represent K number of centers which are: $C_y = \{c_1, c_2, \cdots, c_K\}$
Step 2: Assign other replicas to C_y by applying the following steps:
for $(i = 1, i \leq |ARS|, i++)$
for $(k = 1, k < K, k++)$
for $(j = 1, j = N, j++)$ Compute $L(\otimes S_{ij}, \otimes c_{kj})$ using Equation 5.13
Assign S_{ij} to C_p iff $L(S_{i,j}, c_p)$ is the minimum , where, $p = \{1, 2, \cdots, K\}$
for j end
for k end
$C_i = c_p$
for i end
Step 3: Compute new centers of clusters: $\{c^*_1, c^*_2, \cdots, c^*_K\}$
where $c^*_l = \frac{1}{n_y} \sum_{x_l \in C_y}((\bar{x}_{ij}), (\underline{x}_{ij}))$, where n_y is the number of points in C_y
If $c^*_y = c_y$ then terminate. Otherwise go to *Step 2*
Step 4: Return $DGT = GIT + C_i$
End

$No.$	Attribute symbol	Attribute name
1	a_1	Quality of availability
2	a_2	Quality of Security
3	a_3	Price respectively
4	a_4	Response time
5	a_5	Frequency visited
6	a_6	Type of file
7	a_7	Bandwidth
8	a_8	Cost per MB
9	a_9	Reliability
10	a_{10}	Number of streams

Table 5.6: List of attributes

S_i	a_1	a_2	a_3	a_4	a_5	a_6	D_i
S_1	VG	G	G	G	G	G	1
S_2	MG	MG	F	G	P	F	0
S_3	F	MP	MG	G	G	MG	0
S_4	MG	MP	G	G	G	G	1
S_5	F	P	G	MG	P	G	1
S_6	MG	MP	F	G	MP	F	0
S_7	G	G	G	G	G	MG	1
S_8	G	P	G	P	P	G	0
S_9	MG	MP	G	G	G	G	1
S_{10}	G	P	G	P	G	F	1

Table 5.7: Linguistic attributes values: Category Table

S_i	a_1	a_2	a_3	a_4	D_i
S_1	[9.25,9.90]	[7.50,8.50]	[6.50,7.50]	[7.50,8.00]	1
S_2	[5.00,6.00]	[5.50,6.00]	[4.75,5.00]	[6.75,8.25]	0
S_3	[4.75,5.00]	[3.25,4.00]	[5.25,6.00]	[7.50,9.00]	0
S_4	[5.50,5.50]	[3.00,3.50]	[6.50,7.50]	[7.75,8.25]	1
S_5	[4.50,5.00]	[2.50,3.00]	[6.50,8.50]	[5.25,6.00]	1
S_6	[5.25,6.00]	[3.00,3.50]	[4.25,5.00]	[7.50,9.00]	0
S_7	[8.70,9.00]	[6.50,7.50]	[7.50,7.50]	[7.50,8.00]	1
S_8	[8.50,9.00]	[2.30,3.00]	[6.50,8.50]	[1.25,3.00]	0
S_9	[5.50,5.50]	[3.00,3.50]	[6.50,7.50]	[7.75,8.25]	1
S_{10}	[8.50,9.00]	[2.30,3.00]	[6.50,8.50]	[1.25,3.00]	1

Table 5.8: Grey Decision table GDT

S_i	a_1^*	a_2^*	a_3^*	a_4^*	D_i
S_1	[0.9343,1.0000]	[0.8823,1.00000]	[0.6538,0.5666]	[0.16666,0.15625]	1
S_2	[0.5050,0.6060]	[0.64705,0.7058]	[0.8947,0.8500]	[0.18518,0.15151]	0
S_3	[0.4797,0.5050]	[0.38235,0.4705]	[0.8095,0.7083]	[0.16666,0.13888]	0
S_4	[0.5555,0.5555]	[0.35294,0.4117]	[0.6538,0.5666]	[0.16120,0.15151]	1
S_5	[0.4545,0.5050]	[0.29411,0.3529]	[0.6538,0.5000]	[0.23800,0.20830]	1
S_6	[0.5303,0.6060]	[0.35294,0.4117]	[[1.000,0.8500]	[0.16666,0.13888]	0
S_7	[0.8787,0.9090]	[0.76470,0.8823]	[0.5666,0.5666]	[0.16660,0.15620]	1
S_8	[0.8585,0.9090]	[0.29411,0.3529]	[0.6538,0.5000]	[1.00000,0.41666]	0
S_9	[0.5555,0.5555]	[0.35294,0.4117]	[0.6538,0.5666]	[0.16120,0.15150]	1
S_{10}	[0.8585,0.9000]	[0.29411,0.3529]	[0.6538,0.5000]	[1.00000,0.41666]	1

Table 5.9: Grey Decision Table (GDT) using the normalization Equations

$L(S_0, S_i)$	a_1^*	a_2^*	a_3^*	a_4^*
$L(S_0, S_1)$	0	0	0.4473256789	0.8730757612
$L(S_0, S_4)$	0.5839615752	0.7913896498	0.4473256789	0.8796244932
$L(S_0, S_5)$	0.6893338117	0.8744746322	0.4922626181	0.7898744482
$L(S_0, S_7)$	0.1065405208	0.1663780662	0.5177408189	0.8730757612
$L(S_0, S_8)$	0.1183371163	0.8744746322	0.4922626181	0

Table 5.10: Distance between the Ideal Provider and positive region when $D_i = \backslash yes$"

$L(S_0, S_i)$	a_1^*	a_2^*	a_3^*	a_4^*
$L(S_0, S_2)$	0.5826496934	0.3766543669	0.1052631579	0.8568713488
$L(S_0, S_3)$	0.6720019143	0.7282010825	0.2373828629	0.8784104612
$L(S_0, S_6)$	0.5643021302	0.7913896498	0 .000000000	0.8784104612

Table 5.11: Distance between the Ideal Provider and positive region when $D_i = \backslash no$"

$\Gamma(S_0, S_i)$	Comparative sequences
$\Gamma(S_0, S_1)$	0.6247258204
$\Gamma(S_0, S_2)$	0.4539037446
$\Gamma(S_0, S_3)$	0.2849231857
$\Gamma(S_0, S_4)$	0.2319730128
$\Gamma(S_0, S_5)$	0.1911476053
$\Gamma(S_0, S_6)$	0.3650409183
$\Gamma(S_0, S_7)$	0.5271461913
$\Gamma(S_0, S_8)$	0.5779237679

Table 5.12: Calculation of GRC between positive region and the Ideal Provider (Equation 5.18)

$\Gamma(S_0, S_i)$	BRS
$\Gamma(S_0, S_1)$	0.6247258204
$\Gamma(S_0, S_8)$	0.5779237679
$\Gamma(S_0, S_7)$	0.5271461913
$\Gamma(S_0, S_2)$	0.4539037446
$\Gamma(S_0, S_6)$	0.3650409183
$\Gamma(S_0, S_3)$	0.2849231857
$\Gamma(S_0, S_4)$	0.2319730128
$\Gamma(S_0, S_5)$	0.1911476053

Table 5.13: Best Replicas Set

Sites	A	S	T	C
S_1	60	75	50	40
S_2	60	75	50	40

Table 5.14: Comparison 1 between GRST and K-means approach

Sites	A	S	T	C
S_1	50	50	99	99
S_2	99	99	50	50

Table 5.15: Comparison 2 between GRST and K-means approach

Sites	A	S	T	C
S_1	50	60	99	99
S_2	99	99	50	50

Table 5.16: Comparison 3 between GRST and K-means approach

S_i	a_1^*	a_2^*	a_3^*	a_4^*	C_y
S_1	[0.93,1.00]	[0.88,1.00]	[0.77,0.67]	[0.40,0.40]	1
S_2	[0.51,0.61]	[0.65,0.71]	[1.00,1.00]	[0.44,0.44]	2
S_3	[0.48,0.51]	[0.38,0.47]	[0.95,0.83]	[0.17,0.13]	2
S_4	[0.57,0.57]	[0.35,0.41]	[0.77,0.67]	[0.39,0.39]	1
S_5	[0.45,0.51]	[0.29,0.35]	[0.77,0.59]	[0.57,0.57]	1
S_6	[0.53,0.61]	[0.35,0.41]	[1.00,1.00]	[0.40,0.40]	2
S_7	[0.88,0.91]	[0.76,0.88]	[0.67,0.67]	[0.40,0.40]	1
S_8	[0.86,0.91]	[0.29,0.35]	[0.77,0.59]	[1.00,1.00]	2
S_9	[0.56,0.56]	[0.35,0.41]	[0.77,0.67]	[0.39,0.39]	1
S_{10}	[0.86,0.91]	[0.29,0.35]	[0.77,0.59]	[1.00,1.00]	1

Table 5.17: Decision table with normalized grey attributes of K-GRST

Chapter VI

Rough set based Quality of Service Design for service provisioning in Clouds

In this chapter we explain how the selection strategy of the Fine-grain phase can be extended for use in the resource broker of Cloud environment [GWAR11].

6.1 Introduction

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [clo]. By dynamically provisioning of resources enables cloud computing infrastructure to meet arbitrary varying resource and service requirements of cloud customer applications.

The application requirements can be characterized by quality of service (QoS) requirements such as availability, security, reliability etc., as mentioned in the Service Level Agreement (SLA). SLA is a legal binding contract which states QoS guarantees that an execution environment (provider) agrees to provide its hosted application with. The Cloud Computing paradigm is composed of three service models namely Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS model enables the customer to use the provider application hosted on the cloud infrastructure. In this model the customer does not have any control over the underlying Cloud infrastructure but has little control over application configuration settings. PaaS model provides the capacity to the consumer to deploy the customer created applications onto the Cloud Infrastructure using the programming language or tools supported by the Cloud Provider.

The Consumer does not manage the underlying cloud infrastructure such as network, storage etc. but has control over the deployed applications. IaaS model allows the customer to provision processing, storage, networks and other important software where a customer can deploy software such as operating system and applications. The consumer does not have control over the underlying infrastructure but has control over operating systems, storage, deployed applications. A cloud environment is characterized by multiple providers, each with its own service terms management systems, operating platforms and security levels. The specialized expertise and value addition of a cloud service broker helps IT managers find the right cloud offering, deploy their application in the cloud and manage it properly. Cloud brokers negotiate the best deals and relationships between cloud consumers and cloud providers. They can use specialized tools to identify the most appropriate cloud resource and map the requirements of application to it. They can also be dynamic by automatically routing data, applications and infrastructure needs based on some QoS criteria like availability, reliability, latency, price etc.

Cloud broker services are mainly classified into three categories viz. Service Intermediation, Service Aggregation, and Service Arbitrage. Service Intermediation broker provides a service to a consumer that enhances a given service by adding some value on top to increase some specific capability. Service Aggregation brokerage service combines and integrates into one or more services and ensures data are modeled across all component services and movement and security of data between the service consumer and multiple providers. Service Arbitrage is similar to cloud service aggregation but services being aggregated are not fixed. In addition these services provide flexibility and opportunistic choices for the service aggregator.

In our work, a Modified rough set based Cloud-QoS Management Strategy (MC-QoSMS) framework is thesis to be employed in a cloud broker of cloud environment [clo]. It is to discover the best service provider among all the available service providers in minimum searching time. The term "best" means, the elements of all QoS parameters in the provider match the requirements of the cloud user/ application.

The concept of reduct from rough set theory is used to allocate the best service provider to the cloud's user with minimum searching time. The performance of the proposed system has been analyzed in terms of number of requests. It is reported that the system outperformed random algorithm by 25% and the round robin algorithm by 30% for 100 requests.

6.2 Background

The computational capacity of the local resources can be increased by hiring IaaS provider. In literature, two market oriented scheduling policies to optimize time and cost have been proposed by M.A.Salehi and Rajkkumar Buyya [HYPY10]. In the present framework, the searching time is optimized by using a concept called reduct of rough set theory [Sko85].

The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same predictive capability of the decision feature as the original. A reduct is defined as a subset of minimal cardinality R_{min} of the conditional attribute set C such that $\gamma_R(D) = \gamma_C(D)$, where γ represents dependency of attributes, C and D represent conditional and decision attributes of given information system respectively. Reduct can be

$$R = \{X : X \subseteq C;\ \gamma_X(D) = \gamma_C(D)\} \tag{6.1}$$

$$R_{min} = \{X : X \epsilon R;\ \forall Y \epsilon R;\ \mid X \mid \leq \mid Y \mid\} \tag{6.2}$$

6.3 System Overview

In this section we present an Architecture of MC-QoSMS framework

6.3.1 Architecture of MC-QoSMS Framework

QoS parameters used in MC-QoSMS framework can be classified into three layers namely, Application layer QoS (ALQoS), Middleware Layer QoS (MLQoS) and Netware Layer Qos (NLQoS). ALQoS parameters include Accessibility, Security and

Availability; MLQoS parameters include frequency of CPU, secondary memory storage, cluster and GPU while NLQoS parameters include network availability, bandwidth, latency and error rate. In the present model, various parameters namely Availability, Security, Processor frequency, Main Memory Storage, Secondary Memory Storage, GPU, I/O performance and Network Round Trip Time(RTT) of the three QoS layers have been considered while all other parameters mentioned in the Service Level Agreement (SLA) are assumed to be met by the provider.

The MC-QoSMS framework has been divided into four components as shown in figure 6.1 Cloud Registry(CR) is registry service used to record service capability and QoS provisions for different cloud managers by different service providers. Cloud Manager (CM) is used to host the application of the client by a service provider. *Eucalyptus* [NWG+09] environment may be considered by a service provider for service and execution environment with QoS specifications for IaaS. Network Resource Manager (NRM) uses the concept called Bandwidth Broker which is responsible for managing the communication between the user and the cloud manager. Application QoS Manager (AQoSM) has a specific strategy using MC-QoSMS algorithm to find service providers adhering QoS requirements as specified for a service in an SLA. When a service provider has been selected, the AQoSM coordinates with the service providers cloud manager for subsequent service execution.

Cloud providers publish their service along with all types of QoS parameters in the CR then MC-QoSMS algorithm will be invoked. The MC-QoSMS algorithm proposed in the present proposal is given in the next section.

6.3.2 MC-QoSMS Algorithm

In this section an algorithm of the proposed approach MC-QoSMS 6.1 is explained

6.3.3 Interaction with NRM

NRM is used to monitor the network links between the users and the providers. AQoSM interacts with NRM at two different occasions:

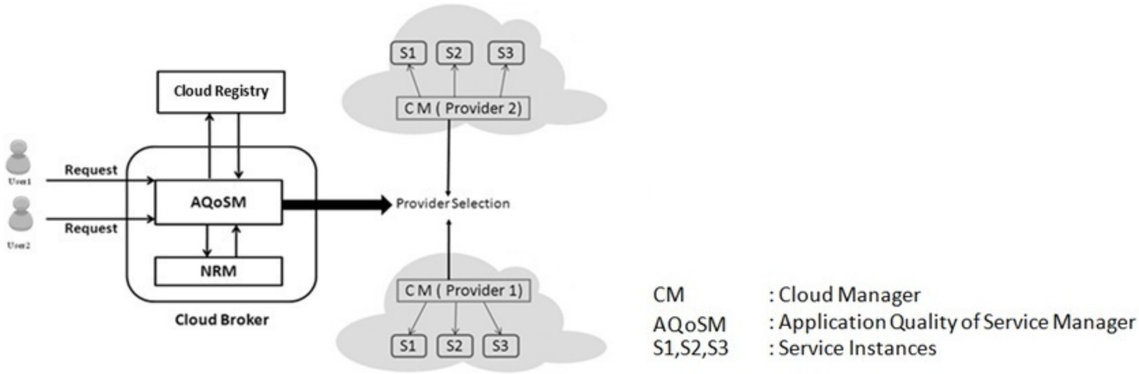

Figure 6.1: MC-QoSMS Framework [GWAR11]

1. AQoSM after applying reducts gives minimized set of providers. Then AQoSM interacts with NRM to monitor the reduced set of provider's network links which closely matches with Network QoS parameters as specified in the SLA. NRM then gives back provider's RTT to AQoSM.

2. Once an appropriate service has been discovered by the strategy used by AQoSM then it interacts with NRM for provisioning of network resources monitoring and supporting admission control. The NRM determines a communicated path between user and the destination. Then the Network path gets reserved using protocols like RSVP as per the Network QoS parameters specified in the SLA for the service. The sequence diagram of the C-QoSMS framework is shown in Figure 6.2.

There is no single universally accepted standardization available in the literature for QoS parameters. In our work we have used Amazon's various MLQoS parameters standardized on 10 point scale [ama]. The details of Standardization of QoS parameters of various applications selected by the client are given in [GWAR11]. One of the applications selected by the client will be processed against the list of providers that the cloud broker has and maps a cloud provider which closely matches the QoS requirements as given by Amazon.

6.4 Results and Discussion

Cloud Providers with varying performance in main memory, processor, storage, availability, and RTT for a database application [GWAR11] (IaaS Service Model) have

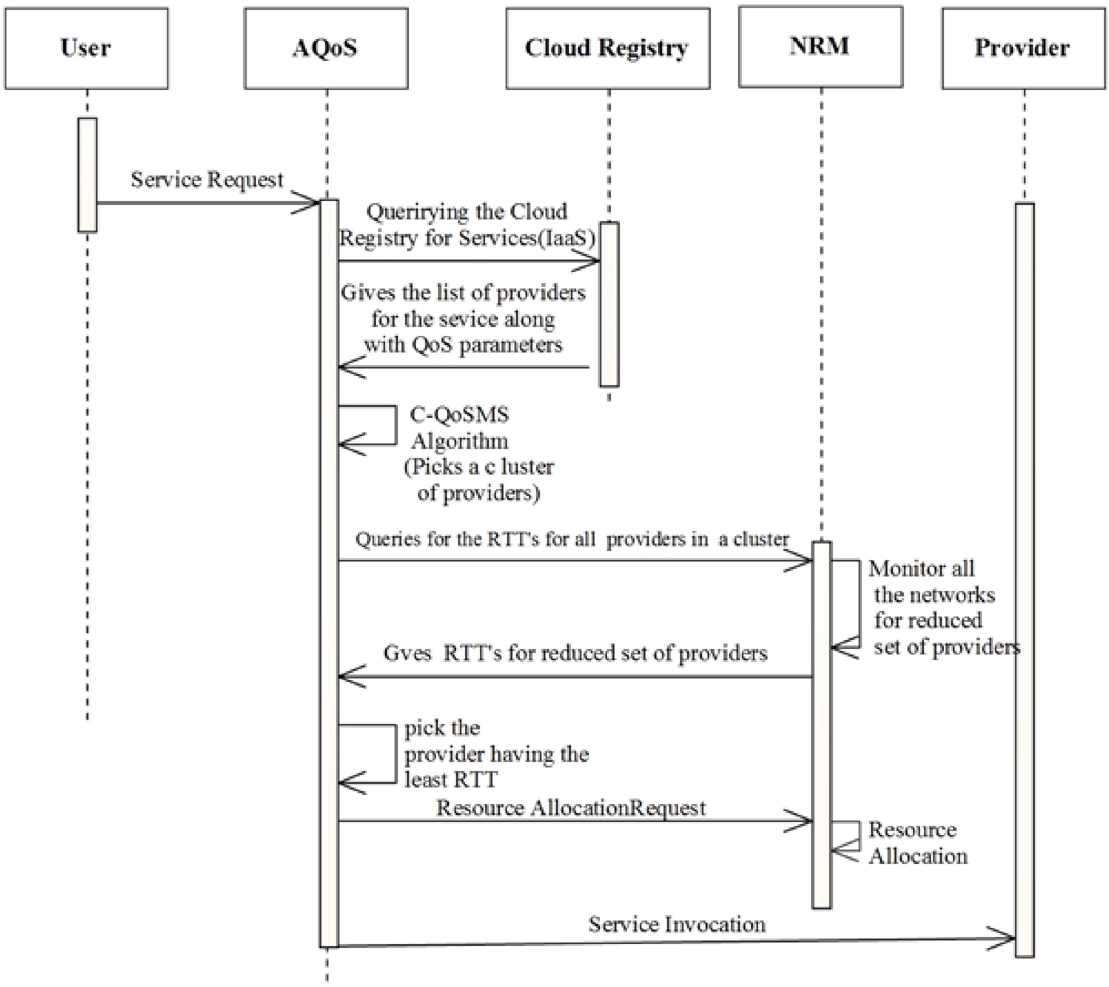

Figure 6.2: Sequence diagram of MC-QoSMS Framework [GWAR11]

Algorithm 6.1 Main steps of MC-QoSMS Algorithm

Input:

[1] List of available services (Applications).

[2] User request UR. /* A specific application is chosen by the user*/

Output:

[1] Service provider ID.

Begin

Step 1: Determine the QoS parameters (PA) and their values (PV) of ALQoS and MLQoS for the user request. /* For instance Amazon classified QoS parameters into six types of applications [ama]*/

Step 2: Contact the cloud registry by cloud brokerage to get QoS specifications all IaaS providers.

Step 3: Form Information System (IS) table.

Step 4: Determine decision attribute using a clustering process.

Step 5: Generate reducts of Decision Information System (DIS) table.

Step 6: store the generated Reducts in Reducts Repository (RRP).

Step 7: Converts user requirements to the QoS metrics, such as deadline work time, and required cost using Resource requirement interpreter.

Step 8: Select a reduct with closest QoS parameters to the user request (required QoS of user's request).

Step 9: An equivalence class of a reduct is selected which is close to the user's values requirements.

Step 10: Sort Ascending providers of a selected class relative to RTT (NLQoS parameter) metric.

Step 11: Select first provider's Id to serve user request.

End

been studied and analyzed as a function of number of provider requests by measuring Euclidean distance between the ideal and provider's QoS parameters of an application. The present results of the proposed MC-QoSMS framework have been compared with the results obtained using Random algorithm and Round robin algorithm for the same data set. The performance of the proposed framework is found to be closer to the ideal provider and gave best results compared to other providers. The performances of the three algorithms are shown in Figure 7.6. Figure 7.6 shows the performance of the three systems with 20 requests.

The results obtained from the MC-QoSMS framework were found to be ideal provider consistently, and displayed a trend to reach it. The ideal model value is the zero line (the x-axis) which indicates main memory, processor frequency, storage, availability was maximized and round trip time was minimized. The MC-QoSMS selects the best provider at any given time as shown in Figure 7.6, where the line represented by the present system is consistently very close to the x-axis. The present

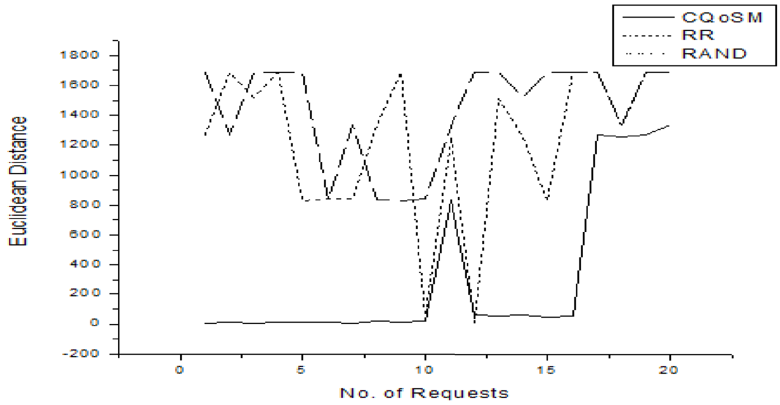

Figure 6.3: Performance of MC-QoSMS, Round Robin and Random algorithms for 20 requests

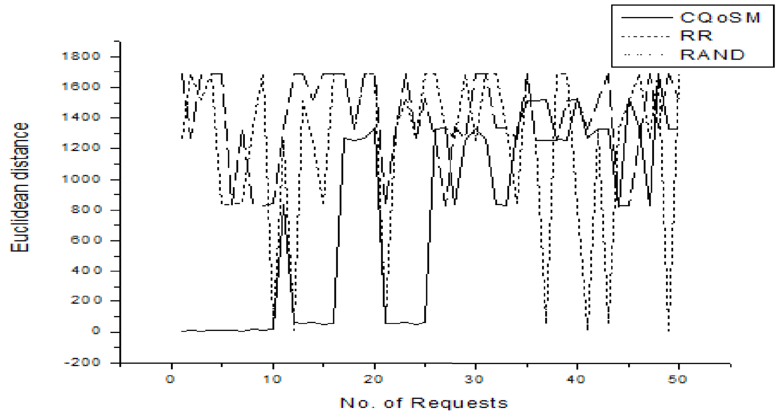

Figure 6.4: Performance of MC-QoSMS, Round Robin and Random algorithms for 50 requests

system selects best provider out of available ones at any point of time. The chances of selecting the best provider may or may not appear to decrease for subsequent requests. However, the system allows other providers to recover or finish processing of some executing jobs and release more system resources. Thus, it will enhance chances of selecting best provider for the next request.

The ability of MC-QoSMS framework has an added value, in contrast with the other two systems namely, RR and RAND methods. The provider selected by these systems (RR and RAND) is overloaded or is not a best provider, as they have high Euclidean Distance(ED)(shown in Figures(7.6, 6.4, 6.5) that indicates bad performance. Instead if allowed to select best provider by the MC-QoSMS, it completes the task and releases resources thereby increasing system resources. The very low values of ED presented by Random and Round Robin algorithms unfortunately occurred

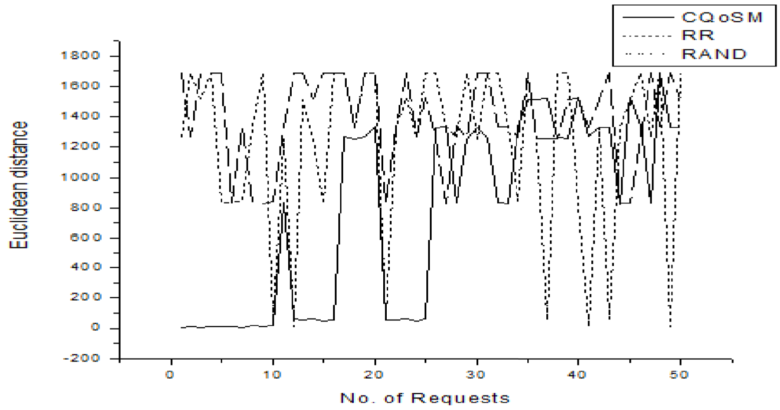

Figure 6.5: Performance of MC-QoSMS, Round Robin and Random algorithms for 100 requests

randomly in an inefficient and unbalanced manner. On the other hand, the stable and managed behavior of the present MC-QoSMS framework ensures a higher level of Main memory, Storage, Processor Frequency and availability.

Figures 6.4 and 6.5 show for 50 and 100 requests respectively. These figures show MC-QoSMS framework still functions properly, even after increasing the number of provider requests in contrast to the other systems. In the study it was also noticed that Random algorithms showed better performance than Round Robin method even after increasing the number of requests. This observation is most likely due to its nature of requesting from each provider in a Random manner. However, the MC-QoSMS framework still performed better than both the other systems. Comparison Figures show that MC-QoSMS framework is more stable than other systems and tends to move closer to the model ideal value (the Zero line). The Average values of Euclidean Distance (AED) for different number of requests obtained in the present frame work are given in Table 6.1.

In order to demonstrate further, the Efficiency(η) of the M-QoSMS framework over other systems is given by Equation 6.3.

$$\eta = \frac{AVOS - AVUS}{AVOS} \times 100 \tag{6.3}$$

In Equation 6.3, AVOS represents Average value of the Euclidean distance of the other systems and AVUS stands for Average value of Euclidean distance of the

Table 6.1: Average values of Euclidean Distance and η(%) for different no. of requests

System/Framework	No. of requests	AED	η (%)
M-QoSMS	20	318.08	-
	50	766.92	-
	100	928.79	-
Random	20	1151.00	72.36
	50	1223.36	37.31
	100	1248.07	25.58
Round Robin	20	1248.08	77.84
	50	1390.91	44.86
	100	1338.13	30.59

underlying system. Using the AED, the value of η has been calculated for different number of requests and is included in Table 6.1. It can be seen that the efficiency of MC-QoSMS framework over random is 25.58% and round robin is 30.59% for 100 requests that seems to be closer to the MC-QoSMS framework than the round robin algorithm in the present study.

6.5 Conclusions

To meet the requirements of both cloud users and service providers, effective and efficient resource broker is proposed with new MC-QoSMS framework. Rough Set Theory is used to minimize number of attributes and minimize searching space by selecting a cluster as required by the user. The proposed framework is more stable than RR and RAND algorithm. The Euclidean distance tends to move to the model ideal value (zero line) indicating enhanced system performance. New QoS parameters namely Main Memory, Processor Frequency, Secondary Storage, Availability, Security and RTT were considered in the present system for a database application in IaaS. The MC-QoSMS framework can be of much benefit to other cloud services that require selection of service provider in cloud environment, utilizing less time and demonstrating high quality of performance.

6.6 Chapter Summary

In this chapter we have started with some essential concepts related to services in cloud environment. then the background of this work is briefly given. Then we have presented a system overview the architecture of MC-QoSMS Framework. This is followed by MC-QoSMS algorithm and interaction with NRM. We take up then actions to be taken by a broker daemon for the replica selection. A generic use case is presented for such a broker. Then we compare our approach with others. Finally a conclusion is given.

Chapter VII

Thesis Summary

In this work, the design and implementation of high level brokering service that can be used with data grids is discussed. Broker uses information regarding replica location and user preferences to guide selection from among storage replica alternatives in order to ensure reliable bulk data transfer with constrained time requirements. Extracting multiple replica providers that are stable and having a close match to the user's requirements is one solution to this problem. To reach this objective, a two-phased service oriented broker has been proposed with two phases of selection policies to reduce transfer time costs by discovering, selecting, reserving and assigning the best associated providers.

7.1 Thesis Focus

The strategies proposed in the thesis reduce the overheads of the selection mechanisms that drive the data management components of a data grid by extending the resource broker with policies that factor in user quality of service as well as service time cost when transferring data. The main objective is to design a scalable service oriented broker for the selection process that will locate best set of data servers (replica providers) when large data files movement is required by the data grid job. The service oriented broker operates within data grid management framework.

The proposed broker improves the selection service from the individual replica data transfer site into set of replica sites transfers, by factoring not just explicit cost of transfer (counting number of Hops between computing element and replica providers), but it can, through utilize the core services, of the network monitoring services mechanism such as *Network Weather (NWS)* and *Iperf*, improve the selection criteria depends on the network traffic and increase user received quality of service with a sustainable set of data resources. Furthermore, the work will be applicable to all large volume transfers common in the *LHC*.

The ideas are inspired from real world where the players involved, the producers and the consumers are assumed to be rational entities that is consumers base their purchasing decisions intelligently from quantitative and qualitative analysis of the quality of the service they receive and the producers register their services according to factors such as availability, security, cost per MB and etc. The service oriented broker aims to:

1. Improve the performance and ensure scalability for large-scale transaction volumes
2. Deal with the vagueness and uncertainty between advertised and requested resources
3. Transfer large amount of data (Terabyte or above) from different sources simultaneously
4. Allow Data consumers are to get portions of data from different locations
5. Works properly with dynamic and static replica strategies
6. Match best *QoS* between user's request and data providers

7.2 Concluding Remarks

Time and cost are the most important factors in most selection processes in the Data Grids. In the thesis a scalable replica selection broker is proposed with dynamic replica selection strategies. The novelty of these strategies is that it aims to deal with a group of plausible and stable replica providers at the same time.

The process of accessing different parts of the required data simultaneously from different places is one of the properties of transport service such as GridFTP. To utilize partitioning feature of GridFTP, the replica selection technique should select more than one providers at a time when the request was made. However, to decrease the total file transfer time, selected providers should have stable links (uncongested). In order to achieve this goal an association technique of Data Mining approach is needed.

Generally, the replica providers have different properties, and users/applications have changing demands. To provide the best quality of service to the user another selection strategy should be added. This selection strategy is applied on the set of the stable associated replica providers. It extracts another set of providers who have similar properties close to the user's requirements. This is a multi attribute decision making problem. To solve this problem the Grey-based Rough Set theory is used to work with both types of attributes, numeric and non numeric. For both selection techniques, a high level service broker is designed in the Data Grid environment.

The proposed research work provides a design of a novel broker for Data Grid infrastructure for the management of storage resources and data that are distributed across Grid environments.

We construct a higher-level service on top of layered set of core services, such as data transport, security, and replica cataloging. In this thesis, we discuss the design and implementation of a high-level replica selection service called 'Two phased Service Oriented Broker' that uses information regarding replica location and user preferences to guide selection from among storage replica provider alternatives. We first present a basic replica selection service design, then show how dynamic information collected from RLS and NMS information service capabilities concerning storage system properties can help in improving and optimizing the selection process. The features of 2SOB are, scalable, reliable, efficient and easy for deployment.

EU Data Grid data is used as a test data to validate the presented approach; first phase has been applied on some real network data (CERN, Date: February, 2011). The result shows an improvement in selection using the new strategy compare to previous selection methods of different brokers.

The first selection strategy (Coarse-grain phase) has been applied on the real network data of (February, 2011) from CERN. The second strategy (Fine-grain phase) has been applied as a model with synthetic data based upon the storage application in the cloud environment of *Amazon*. Experimentation results are compared with other contemporary selection methods of different brokers that showed better results. Using the proposed broker it was possible to achieve an enhancement in the speed of executing job through reducing the transfer time. Before deploying replica placement and selection algorithms in real Grid environment the algorithms must be tested thoroughly. We evaluate the performance and feasibility of the replication algorithms by using a simulation toolkit, OptorSim V 2.0 designed for Data Grids [Gric].

7.3 Performance and Results

This section describes the performance of our proposed selection strategies in the 2SOB.

7.3.1 Performance and Results of Coarse-grain phase

The first phase, known as Coarse-grain has been applied on real network data (February, 2011) obtained from CERN [cot]. Comparisons were made with other contemporary selection methods of different brokers and also with various replica selection methods that were proposed in the literature. Tests have been performed in two different selection scenarios:

1. Selection of a single replica provider and,
2. Selection of multi-replica providers

In our experiments, we have used nine jobs, each requests a single file. The sizes of files are (1, 1.2, 1.4, 1.7, 2, 3, 4, 5, 6)GB. The files were replicated in 30 replica providers in different locations. Six selection methods (including ours) have been used in our experiments which are: (TM(BW), TM(HopCount) NN, KNN, GA, EST(proposed)). Also some selection methods such as KNN and GA have been tested with both of their possibilities:

- correct classification (predicting) (the required data was available in selected provider), and

- with the wrong classification (the required data was not available in the selected provider)

The results of $Av(\tau)$ (represents time of running the selection method plus time of transmission the file from the selected provider to the computing site) are tabulated according to the following steps:

1. Each job is executed using all six selection methods to select best replica provider
2. Run time (posteriori analysis) of each selection method is recorded
3. Transfer times from the selected providers to the computing site are recorded
4. Total time (τ) is computed for all methods separately
5. Steps (1-3) are repeated for other nine jobs (each job has a single file)
6. Average time $Av(\tau)$ is computed for each selection method

The values resulting from the simulation are tabulated in Tables 7.1 and 7.2

Furthermore, in order to demonstrate the efficiency of the EST strategy over the others, Equation 7.1 is used.

$$\eta = \frac{AVOS - AVUS}{AVOS} \times 100 \tag{7.1}$$

Where, $AVOS$ represents Average distance value of the other strategies and $AVUS$ stands for Average distance value of the our strategy. Same procedure is repeated when multi-providers are selected. The results of $Av(\tau)$ and η for both scenarios are formed in Table 7.1 and Table 7.2.

Table 7.1: Comparison between EST and others using a single replica provider

$Selection\ Methods$	$Av(\tau)$	η (%)
$EST(proposed)$	85.444	-
$TM(BW)$	113.55	24.38
$TM(HopCount)$	122.00	30.28
$KNN(C)$	114.45	25.34
$KNN(W)$	136.55	37.17
NN	126.66	32.54
$GA(W)$	145.70	41.34
$GA(C)$	102.44	16.59

Table 7.2: Comparison between EST and others using two replica providers

Selection Methods	$Av(\tau)$	η (%)
$EST(proposed)$	64.78	-
$TM(BW)$	102.2	36.63
$TM(HopCount)$	95.44	32.13
$KNN(C)$	108.3	40.21
$KNN(W)$	135.4	52.17
NN	114.4	43.40
$GA(C)$	82.44	21.43
$GA(W)$	93.00	30.34

7.3.1.1 Analysis of results of Coarse-grain phase

For comparing two methods at a time Statistical Paired T-test is performed with the following conclusions:

1. For K methods we have K_{C_2} possible test cases
2. Incident matrix is formed with values 0, if comparisons are statistically significant or 1 if insignificant respectively
3. This matrix is used to form Connected Components
4. We observed that with the 5 methods we compared, our proposed method outperformed all others in this phase, as our method forms connected component with a singleton element with least mean of file transfer time
5. Our method is shown to be consistent and more stable than others
6. Stability of the method depends on the size and state of the Grid, that may not be under our control all the time

To properly analyze the results of selecting single and multiple replica providers using six methods we draw Figure 7.1 and Figure 7.3. These figures show that the average time, $Av(\tau)$ of the proposed strategy is less compared with other selection strategies. Moreover to find the difference between the efficiency of our strategy and the others we draw the Figure 7.2 for single replica provider selection, and Figure 7.4 for multiple replica provider selection. The result shows that the performance of the proposed strategy EST is better compare to other selection strategies. Followings are the reasons behind this:

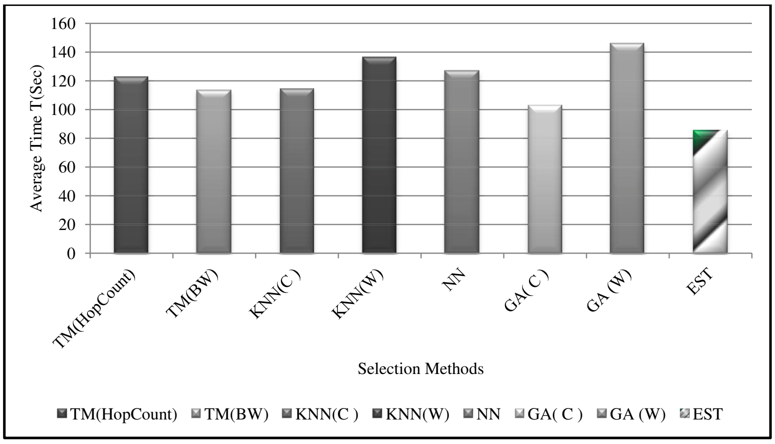

Figure 7.1: Comparison between EST and other selection strategies when a single provider is selected

- **Availability:** In our proposed strategy, EST there is no possibility of unavailability of the file in the selected provider since we do not avoid catalog lookup time
- **Reliability:** EST selects set of providers at a time, so in case when the selected provider fails another provider takes place
- **Stability:** EST selects providers which have stable network links, i.e., less percentage of lost packets of the required data
- **Efficiency:** EST selects a *set* of replica providers which are stable, reliable and available. The required files are simultaneously transferred from the selected set

Figures 7.1 and 7.4, show that the average time of our proposed strategy, EST is less when the number of selected providers increases. As we can see that, the total time in the first case (when a single provider is selected) is about *80 Sec*, and it is about *60 Sec* in case of two providers are selected. In Figures 7.2 and 7.4, the results show that, when the number of providers increases, the efficiency of using EST will also increase, as the effect of the associated stable links of the providers becomes apparent. That means using our proposed broker it is possible to achieve an enhancement in the speed of executing jobs through reducing the transfer time as shown in Figure 7.3.

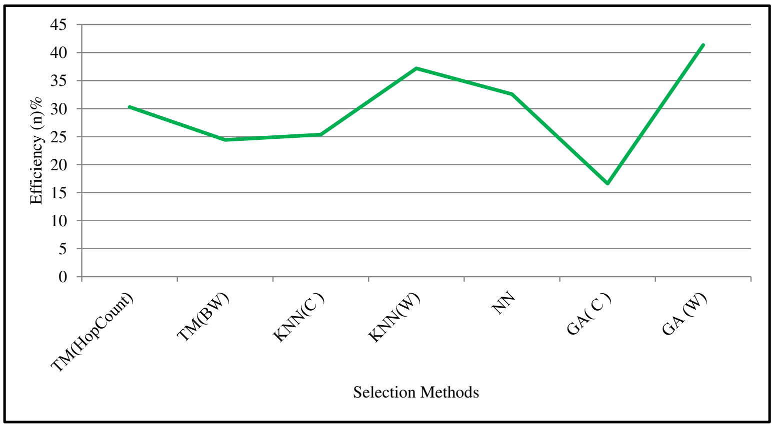

Figure 7.2: Comparison between Efficiency of EST and other selection strategies when a single provider is selected

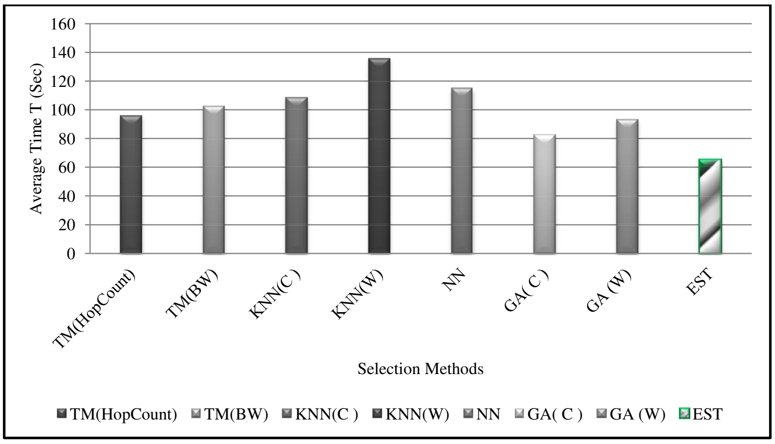

Figure 7.3: Comparison between EST and other selection strategies when two providers are selected

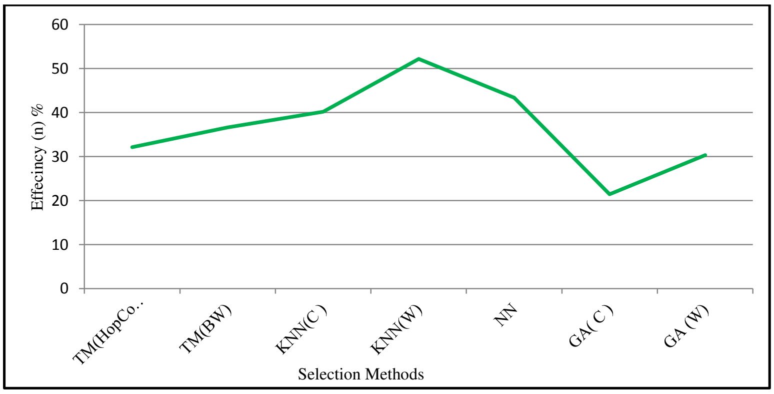

Figure 7.4: Comparison between Efficiency of EST and other selection strategies when two providers are selected

7.3.2 Performance and Results of Fine-grain phase

The strategy of the second phase, known as Fine-grain phase is applied on the synthetic data sets for 100 providers with 10 numeric and non-numeric properties (attributes). We compare our strategy with other replica selection strategies.

The strategy of Fine grain phase (GRST strategy) has two objectives which are:

- *Reducing searching time* of selection process by reducing number of attributes and searching space
- *Accuracy of the selection*; it selects the provider(s) with the closest match to the user/application requirements

The following steps have been used in the experimentation:

1. Forty user requests with different QoS requirements were sent to the five brokers (each one run different selection strategy) simultaneously
2. Each user QoS parameter has a weight associated with it
3. We tested our GRST strategy with four other selection strategies which are (K-means, RGA, Random and Round Robin(RR)) with respect to the user requirements
4. Selection strategy was run to select a single provider using different methods
5. We tabulated the distance between the QoS of user requirements and QoS of the service of the selected providers for all methods
6. The distance between the users QoS demands (required service) are compared with the selected providers of the five selection methods (resulted services)
7. Average distances of all methods are computed as shown in Table 7.3

Equation 7.1 is used to demonstrate the Efficiency(η) of the proposed Grey Rough Set Technique (GRST) compare to other methods.

7.3.2.1 Analysis of results of Fine-grain phase

To test our strategy, GRST, 40 users requests are submitted at the same time to 2SOB. Some of the user's requirements are available and match providers' services. In our simulation, the requests which have indexes from $(1-10)$ and $(20-30)$ have a matched replica providers, whereas the requirements of the requests which have indexes from $(11-19)$ and from $(21-40)$ do not exactly match the available services.

The results show in Figure 7.5 demonstrates that our strategy achieved better performance in term of accuracy and efficiency in the selection.

Table 7.3 shows that the proposed system outperforms random algorithm by 85%, outperforms the round robin algorithm by 80% outperforms the $K-means$ algorithm by 46% and outperforms the rank genetic algorithm by 37%.

The reasons behind that are:

- **Accuracy:** The selected providers using GRST have a closest match to the user's requirements. As it is seen in the figure, the accuracy of selection is high for the first ten request $(1-10)$ where the user's requirements are available. For the next ten requests $(11-19)$, GRST chooses the closest to the user's demand as the user's requirements are unavailable. For requests $(21-30)$ it again meets the exact demand and finally for $(31-40)$ it is closest to the demand compare to all other methods
- **Efficiency:** Using GRST, the selection process becomes more efficient using lower and upper approximations as it reduces the search space of possible service providers

GRST simplifies the search for dominating attributes leading to specific properties, or just rules pending in the data. The results can be summarized as follow:

- Performance index is obtained for all the methods for a set of instances of jobs (40 user request). It is observed that our method's performance index is better than others

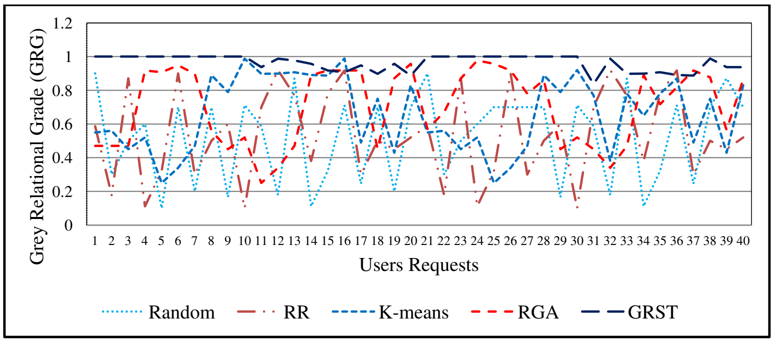

Figure 7.5: Comparison between GRST and other selection strategies

- Our method out performed other methods Statistically when we carried out the analysis of the data using Paired T-test
- The index of Our method is consistent and stable comparative to all other methods
- This indicates that selection of site(s) is robust and does not depend on the file size or type of data sets access

Table 7.3: Comparison between the performance of GRST and others using GRG and $\eta(\%)$ for 40 different requests

System/Framework	$Av(GRG)$	η (%)
GRST	0.96	-
Random	0.52	85.50
Round Robin	0.54	80.01
K-means	0.66	46.84
Rank GA	0.70	37.31

Using another concept of Rough Set Theory (RST) called Reduct, proposed Fine-grain strategy was upgraded to a Modified rough set based Cloud-QoS Management Strategy (MC-QoSMS) that works in Cloud environment. MC-QoSMS framework is proposed to be deployed in a cloud broker of cloud environment [clo]. MC-QoSMS has been applied on the model with synthetic data based upon the storage application in the cloud environment of *Amazon*. Experimentation results compared with some other contemporary selection methods of different brokers showed better results. Using the proposed broker it was possible to achieve an enhancement in the speed of executing job through reducing the transfer time as shown in Figure 7.6 [clo]. It discovers the best service provider among all the available service providers in minimum

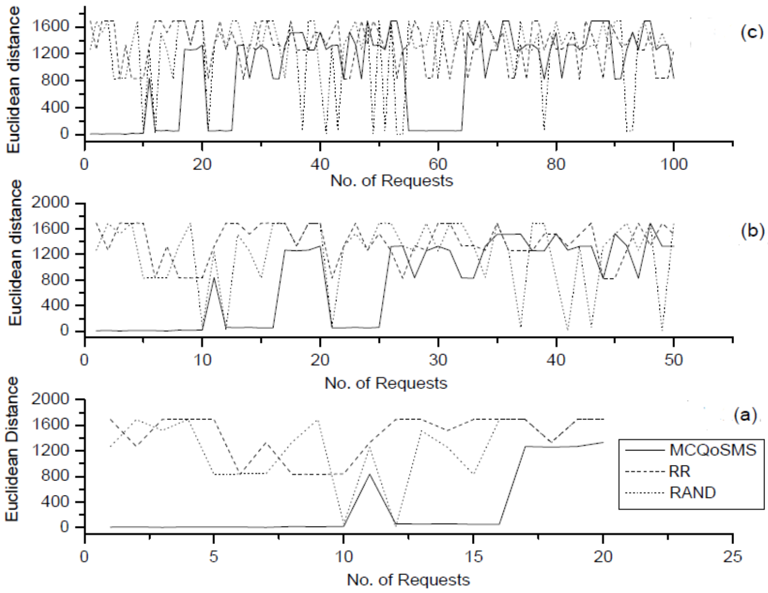

Figure 7.6: Performance of MC-QoSMS, Round Robin and Random algorithms for 20, 50 and 100 of users requests

searching time. The term "best" means, the elements of all QoS parameters in the provider match the requirements of the cloud user/application.

we used Amazon's applications and it's standardized attributes values are used to test the strategy. The test results showed good improvement compared with other strategies. The performance of the proposed strategy has been analyzed in terms of number of requests. It outperformed random algorithm by 25% and the round robin algorithm by 30% for 100 requests as shown in Figure 7.6. Figure 7.6 shows the performance of the three selection strategies with 20, 50 and 100 requests. Our strategy, MC-QoSMS always selects a closest provider to the user/application request.

The figure shows that when the number of requests increases then the performance of the proposed strategy improves. The distance of our strategy is zero when the requirement of user request is available. This indicates that it picks up the provider who has the required service. In case the requirements of user are not available the proposed strategy selects the closest provider. To compare with other strategies such as random and round robin. Equation 7.1 is used and the results are tabulated in Table 7.3.

7.4 Future works

As a future research work we wish to do the following:

- Implement and validate our broker, 2SOB model by deploying it in the real Grid environments, e.g., PRAGMA [pra], WestGrid [Wes] or EUGrid [EM06]. This includes integrating our services with the Replica Location Service and the Globus Toolkit [CQ93]. Finally, the simulation results reported in this work demonstrated the applicability and effectiveness of the proposed methodology; so, one of the next steps in our research is to test the approach proposed in a real grid environment

- As a future scope, it is envisaged that the presented model may use a *One Way Ping Service* (OWPS) [owa] as a user level service with *Ipref* and *NWS* (*Network Monitoring Services*) to enhance the selection of associated replica providers. Using Single Trip Time (STT) instead using RTT gives better performance results (practically not possible to do in real time presently). This is because the trip from the sender to the receiver is important and affects the transfer file time, whereas the trip from the receiver to the sender is not important

- Deploy Modified Cloud QoS Management System (MC-QoSMS) in the real Cloud environments, e.g. Amazon [ama]

References

[AAHJ10] O. Aljadaan, W. Abdulal, M.A. Hameed, and A. Jabas. Enhancing data selection using genetic algorithm. In *Proceedings of the 2010 International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 434–439, 2010.

[Aba04] J. Abawajy. Placement of file replicas in data grid environments. pages 66–73, 2004.

[ABC+07] M. Antonioletti, D. Berry, A. Chervenak, P.r Kunszt, A. Luniewski, S. Laws, and M. Morgan. Ogsa data architecture v0.6.6. technical report, open grid forum. 2007.

[ama] Amazon. http://aws.amazon.com/ec2/.

[AS96] R. Agrawal and J.C. Shafer. Parallel mining of association rules. In *Proceedings of the IEEE Transactions on Knowledge and Data Engineering*, volume 8, pages 962–969, Dec. 1996.

[atl] Atlas at the university of chicago. http://hep.uchicago.edu/atlas/.

[ATS93] R. Agrawal, I. Tomasz, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, SIGMOD '93, pages 207–216, New York, NY, USA, 1993. ACM.

[AWN+10a] R. M. Almuttairi, R. Wankar, A. Negi, C. R. Rao, and M. S. Almahna. New replica selection technique for binding replica sites in data grids. In *Proceedings of 1st International Conference on Energy, Power and Control (EPC-IQ)*, pages 187–194, Dec. 2010.

[AWN+10b] R. M. Almuttairi, R. Wankar, A. Negi, C. R. Rao, and M.S. Almahna. New replica selection technique for binding replica sites in data grids. In *Proceedings of Iraq J. Electrical and Electronic Engineering*, volume 6, pages 186–193, 2010.

[AWNR10a] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao. Intelligent replica selection strategy for data grid. In *Proceedings of the 2010 International Conference on Grid Computing and Applications (GCA'10)*, pages 95–101, 2010.

[AWNR10b] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao. Replica selection in data grids using preconditioning of decision attributes by k-means clustering. In *Proceedings of the Vaagdevi International Conference on Information Technology for Real World Problems*, pages 18–23, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[AWNR10c] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao. Rough set clustering approach to replica selection in data grids. In *Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 1195–1200, 2010.

[AWNR10d] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao. Smart replica selection for data grids using rough set approximations. In *Proceedings of the The International Conference on Computational Intelligence and Communication Networks (CICN2010)*, pages 466–471, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[AWNR11] R. M. Almuttairi, R. Wankar, A. Negi, and C. R. Rao. Enhanced data replication broker. In *Proceedings of the 5th Multi-disciplinary International Workshop On Artificial Intelligence (MIWAI2011)*, pages 286–297, 2011.

[BAG00] R. Buyya, D. Abramson, and J. Giddy. Architecture of a resource management and scheduling system in a global computational grid. In *Proceedings of the High Performance Computing, Asia 2000*, pages 283–289, Beijing, China, May 2000.

[BCC+03] W. H. Bell, D. G. Cameron, L. Capozza, A. P. Millar, K. Stockinger, and F. Zini. Optorsim a grid simulator for studying dynamic data replication strategies. In *Proceedings of the International Journal of High Performance Computing Applications*, 2003.

[BCCS+03] W. H. Bell, D.G. Cameron, R. Carvajal-Schiaffino, A.P. Millar, K. Stockinger, and F. Zini. Evaluation of an economy-based file replication strategy for a data grid. In *Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. Proceedings, CCGrid 2003*, pages 661–668, May 2003.

[buy] Wp2, http://edg-wp2.web.cern.ch/edg-wp2/optimization/ros.html.

[BZ70] R. E. Bellman and L. A. Zadeh. Decision-making in a fuzzy environment. In *Proceedings of the Management Science*, volume 17, pages pp. B141–B164. Institute for Operations Research and the Management Sciences INFORMS, Dec. 1970. Research-article.

[CB02] M. Chetty and R. Buyya. Weaving computational grids: How analogous are they with electrical grids? In *Proceedings of th Computing in Science and Engineering*, volume 4, pages 61–71, Los Alamitos, CA, USA, 2002. IEEE Computer Society.

[CER] Cern, http://www.hpcwire.com/topic/storage/cern.

[CFK+00] A. L. Chervenak, I. T. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. In *Proceedings of the Journal of Network and Computer Applications*, volume 23, pages 187–200, 2000.

[cis] Cisco distributed director. http://www.cisco.com/warp/public/cc/pd/cxsr/dd/index.shtml.

[clo] Cloud computing. http://csrc.nist.gov/groups/sns/cloud-computing/.

[cot] Cottrell. ftp://ftp.slac.stanford.edu/users/cottrell/.

[CQ93] M. J. Clement and M.J. Quinn. Analytical performance prediction on multi computers. In *Proceedings of Supercomputing 1993*, pages 886–894, Nov. 1993.

[CZSS02] M. Carman, F. Zini, L. Serafini, and K. Stockinger. Towards an economy – based optimisation of file access and replication on a data grid. In *Proceedings in Workshop on Agent based Cluster and Grid Computing at 2nd International Symposium on Cluster Computing and the Grid (CCGrid2002)*, pages 286–297, May 2002.

[DBX05] H. Ding, L. Benyoucef, and X. Xie. A simulation optimization methodology for supplier selection problem. In *Proceedings of the International Journal Computer Integrated Manufacturing*, volume 18, pages 210–224, 2005.

[DD09] A. Domenici and F. Donno. Static and dynamic data models for the storage resource manager v2.2. In *Proceedings of the Journal of Grid Computing*, volume 7, pages 115–133, 2009.

[Dir] Cisco Distributed Director. Cisco distributed director. http://www.cisco.com/warp/public/cc/pd/cxsr/dd/index.shtml.

[DJL82] Deng and Ju-Long. Control problems of grey systems. In *Proceedings of Systems & Control Letters*, volume 1, pages 288–294, 1982.

[EM06] AD. Earl and HL. Menken. Supporting the challenge of lhc produced data with scotgrid (ph.d. thesis, university of edinburgh cern). 2006.

[FE09] A.M. Fard and M. Ester. Collaborative mining in multiple social networks data for criminal group discovery. In *Proceedings of the International Conference on Computational Science and Engineering, 2009. CSE '09*, volume 4, pages 582–587, Augest 2009.

[FK01] I. Foster and C. Kesselman. The anatomy of the grid: Enabling. In *Proceedings of the International International Journal of High Performance Computational Applications*, volume 15, pages 200–222, 2001.

[FKTT98] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *Proceedings of the 5th ACM conference on Computer and communications security*, CCS '98, pages 83–92, New York, NY, USA, 1998. ACM.

[FM03] C. Ferdean and M. Makpangou. A scalable replica selection strategy based on flexible contracts. In *Proceedings of the Third IEEE Workshop on Internet Applications, WIAPP 2003*, pages 95–99, June 2003.

[FVS08] I. Foster, S. Vazhkudai, and J. Schopf. Predicting the performance of wide-area data transfers. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing (SC 2008)*, pages 286–297, Nov. 2008.

[GII] http://www.hep.ucl.ac.uk/ytl/grid/gris-01.html.

[gloa] Global grid forum (ggf). http://www.ggf.org.

[glob] Globus. http://www.globus.org/grid-software/computation/csf.php.

[gloc] Open grid services architecture (ogsa). http://www.globus.org/ogsa.

[gria] Gridlab resource management system. online at: http://www.gridlab.org/workpackages/wp-9/.

[grib] Gridway open source meta-scheduler. online at: http://www.globus.org/grid-software/computation/gridway.php.

[Gric] The griphyn project. http://www.griphyn.org.

[GS95a] J. D. Guyton and M. F. Schwartz. Locating nearby copies of replicated internet servers. In *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '95, pages 288–298, New York, NY, USA, 1995. ACM.

[GS95b] J. S. Gwertzman and M. Seltzer. The case for geographical push-caching. In *Proceedings of the 5th Workshop on Hot Topics in Operating Systems, 1995. (HotOS-V)*, pages 51–55, May 1995.

[GW96] A. S. Grimshaw and W. A. Wulf. Legion-a view from 50,000 feet. In *Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing*, pages 89–99, Augest 1996.

[GWAR11] P. Ganghishetti, R. Wankar, R. M. Almuttairi, and C. R. Rao. Rough set based quality of service design for service provisioning in clouds. In *Proceedings of the Sixth International Conference on Rough Sets and Knowledge Technology (RSKT)*, pages 268–273, 2011.

[GWF+94] A. S. Grimshaw, W. A. Wulf, J. C. French, A. C. Weaver, and P. F. R. Jr. Legion, the next logical step toward a nationwide virtual computer. In *Proceedings of the 5th Multi-disciplinary International Workshop On Artificial Intelligence (MIWAI2011)*, 1994.

[HJMS+00] W. Hoschek, J. Jaen-Martinez, A. Samar, H. Stockinger, and K. Stockinger. Data management in an international grid project. In *Proceedings of the International Workshop on Grid Computing (GRID 2000)*, December, 2000.

[Hol00] K. Holtman. Object level replication for physics. In *Proceedings of the 4th Annual Globus Retreat*, Pittsburgh, July, 2000.

[HPY00] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 1–12, New York, NY, USA, 2000. ACM.

[HT07] M. Hofmann and B. Tierney. Development phases of a generic data mining life cycle (dmlc). In *Proceedings of the SETP*, pages 5–11, 2007.

[HYPY10] C-H. Hsu, L. T. Yang, J. H. Park, and S-S. Yeo, editors. *Algorithms and Architectures for Parallel Processing*, volume 6081 of *Lecture Notes in Computer Science*, Busan, Korea, May 2010. Springer.

[ipe] http://sourceforge.net/projects/iperf.

[Jag06] A. Jagatheesan. An introduction to data grid management systems (dgms). In *Proceedings of the SBBD*, page 312, 2006.

[JCSZ07] H. Janicke, A. Cau, F. Siewe, and H. Zedan. Deriving enforcement mechanisms from policies. In *Proceedings of the 8th IEEE International Workshop on Policies for Distributed Systems and Networks, 2007. POLICY '07.*, pages 161–172, June 2007.

[JGR08] A. Jabas, R.M. Garimella, and S. Ramachandram. Proposing an enhanced mobile ad hoc network framework to the open source simulator ns2. In *Mosharaka International Conference on Communications, Computers and Applications, 2008, MIC-CCA 2008*, pages 14– 19, Augest 2008.

[Joe09] B. Joe. Do association rules represent supervised or unsupervised learning. 2009.

[JSA09] A. Jaradat, R. Salleh, and A. Abid. Imitating k-means to enhance data selection. In *Proceedings of the Journal of Applied Sciences 2009*, volume 19, pages 3569–3574, 2009.

[KBM02] K. Krauter, R. Buyya, and M. Maheswaran. A taxonomy and survey of grid resource management systems for distributed computing. In *Proceedings of the Journal of Software: Practice and Experience (SPE)*, volume 32, pages 135–164, 2002.

[KF01] R. Kavitha and I. Foster. Design and evaluation of replication strategies for a high performance data grid. In *Proceedings of the CHEP'01*, pages 106–118, Beijing, 2001.

[KG05] U. Kant and D. Grosu. Double auction protocols for resource allocation in grids. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, volume 1, pages 366–371, Washington, DC, USA, 2005. IEEE Computer Society.

[KMW96] R. T. Kouzes, J. D. Myers, and W. A. Wulf. Collaboratories: Doing science on the internet. In *proceeding of AND nap959*, volume 29, pages 40–46, 1996.

[KR01] J. F. Kurose and K. W. Ross. *Computer networking - A top-down approach featuring the internet*. Addison-Wesley-Longman, 2001.

[KS06] V. Kravtsov and V. Stankovski. Service-based resource brokering for grid-based data mining. In *Proceedings of the International Conference on Grid Computing and Applications*, pages 163–169, 2006.

[LA00] T. Loukopoulos and I. Ahmad. Static and adaptive data replication algorithms for fast information access in large distributed systems, icdcs. In *Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS 2000)*, 2000.

[LAB06] H. Lin, J. Abawajy, and R. Buyya. Economy-based data replication broker. In *Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, e-Science '06*, page 90, 2006.

[Leb99] P. Lebrun. The large hadron collider a megascience project. pages 3–11, Erice, Italy, Augest 1999.

[LFS+06] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, and T. Freeman. A multipolicy authorization framework for grid security. In *Proceedings of the 5th IEEE International Symposium on Network Computing and Applications, 2006, NCA 2006*, pages 269–272, July 2006.

[LGFH95] G. Lindahl, A. Grimshaw, A. Ferrari, and K. Holcomb. Metacomputing - what's in it for me? In *Proceedings of the 5th Multi-disciplinary International Workshop On Artificial Intelligence (MIWAI2011)*, 1995.

[LJ05] R. Lasch and C. G. Janker. Supplier selection and controlling using multivariate analysis. In *Proceedings of the International Journal of Physical Distribution and Logistics Management-2005*, volume 35, pages 409–425. Emerald Group Publishing Limited, 2005.

[LL02] E. Letier and A. V. Lamsweerde. Agent-based tactics for goal-oriented requirements elaboration. In *Proceedings of the 24th International Conference on Software Engineering*, ICSE '02, pages 83–93, New York, NY, USA, 2002. ACM.

[LLM88] M. J. Litzkow, M. Livny, and M. W. Mutka. Condor- a hunter of idle workstations. In *Proceedings of the International Conference on Distributed Computing Systems ICDCS*, pages 104–111, 1988.

[LM05] M. Li and M.A.Baker. The grid core technologies. 2005. Wiley.

[LVZ07] M. Lei, S.V. Vrbsky, and Q. Zijie. Online grid replication optimizers to improve system reliability. In *Proceedings of the IEEE International Parallel and Distributed Processing Symposium, 2007 (IPDPS 2007)*, pages 1–8, 2007.

[LYL+06] G-D. Li, D. Yamaguchi, H-S. Lin, K-L. Wen, and M. Nagai. A grey-based rough set approach to suppliers selection problem. In *Proceedings of the 5th International Conference of Rough Sets and Current Trends in Computing 2006, RSCTC 2006*, volume 4259, pages 487–496. Springer Berlin/ Heidelberg, Kobe, Japan, Nov. 2006.

[Nim] Nimrod/g. online at http://www.csse.monash.edu.au/ davida/nimrod/ni mrodg.htm.

[NVG+05] K. Nadiminti, S. Venugopal, H. Gibbins, T. Ma, and R. Buyya. The gridbus grid service broker and scheduler (v.2.4) user guide. In *online at: http://www.Gridbus.org/broker/2.4/manualv2.4.pdf*, 2005.

[NWG+09] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. In *Proceedings of the Cluster Computing and the Grid CCGRID '09, IEEE Computer Society Washington, USA*, 2009.

[owa] One-way active measurement protocol. http://www.internet2.edu/performance/owamp/.

[pin] Pinger reports. http://www-wanmon.slac.stanford.edu/cgi-wrap/pingtable.pl.

[pra] Bragma. http://goc.pragma-grid.net/pragmadoc.

[Puj01] A. K. Pujari. *Data Mining Techniques*. Universities Press, 2001, Andhra Pradesh, India, 2001.

[RAB08] R. M. Rahman, R. Alhajj, and K. Barker. Replica selection strategies in data grid. In *Proceedings of the Journal of Parallel Distributing and Computing*, volume 68, pages 1561–1574, 2008.

[RBA04] R. M. Rahman, K. Barker, and R. Alhajj. Predicting the performance of gridftp transfers. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)*, 2004.

[RF01] K. Ranganathan and I. Foster. Identifying dynamic replication strategies for a high performance data grid. In *Proceedings of the International Grid Computing Workshop*, pages 286–297, Nov. 2001.

[SAAM09] F. Stagni, A. Arenas, B. Aziz, and F. Martinelli. On usage control in data grids. In *Proceedings of the European Conference on Computer Vision (ECCV) ECCV09*, pages 99–116, 2009.

[SBSV98] M. Sayal, Y. Breitbart, P. Scheuermann, and R. Vingralek. Selection algorithms for replicated web servers. In *Sigmetrics Performance Evaluation Review*, volume 26, pages 44–50, 1998.

[Sko85] A. Skowron, editor. *Computation Theory*, volume 208 of *Lecture Notes in Computer Science*. Springer, Dec. 1985.

[slaa] https://confluence.slac.stanford.edu/display/iepm/pinger.

[slab] http://www-iepm.slac.stanford.edu/cgi-wrap/pingtable.pl.

[sob] Service oriented architecture. http://www.service-architecture.com/web-services/articles/service-oriented-architecture-soa-definition.html.

[Sto01] H. Stockinger. Distributed database management systems and the data grid. In *Proceedings of the 18th IEEE Symposium on Mass Storage Systems and 9th NASA Goddard Conference on Mass Storage Systems and Technologies*, San Diego, April 2001.

[TLC+99] B. Tierney, J. Lee, B. Crowley, M. Holding, J. Hylton, and F. Drake. A network-aware distributed storage cache for data intensive environments. In *Proceedings of the IEEE High Performance Distributed Computing conference (HPDC-8)*, Augest, 1999.

[TM05] C. Tan and K. Mills. Performance characterization of decentralized algorithms for replica selection in distributed object systems. In *Proceedings of the 5th International Workshop on Software and Performance (WOSP 2005)*, pages 257–262, 2005.

[VBR06] S. Venugopal, R. Buyya, and K. Ramamohanarao. A taxonomy of data grids for distributed data sharing, management, and processing. In *Proceedings of the ACM Computing Surveys*, volume 38, New York, NY, USA, June 2006. ACM.

[VS02] S. Vazhkudai and J. Schopf. Using disk throughput data. In *Proceedings of the 3rd International Workshop on Grid Computing in Predictions of End-to-End Grid Transfers*, Baltimore, 2002.

[VS03] S. Vazhkudai and J. Schopf. Using regression techniques to predict large data transfers. In *Proceedings of the International Journal of High Performance Computing Applications (IJHPCA)*, Baltimore, August 2003.

[VSG+07] V. Venturi, F. Stagni, A. Gianoli, A. Ceccanti, and V. Ciaschini. Virtual organization management across middleware boundaries. In *Proceedings of the IEEE International Conference on e-Science and Grid Computing*, pages 545–552, Dec. 2007.

[VTF01a] S. Vazhkudai, S. Tuecke, and I. Foster. The data grid: Towards: an architecture for the distributed management and analysis of large scientific datasets. In *Proceedings of the Journal of Network and Computer Applications*, pages 286–297, 2001.

[VTF01b] S. Vazhkudai, S. Tuecke, and I. T. Foster. Replica selection in the globus data grid. In *Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003. Proceedings, CCGrid 2001*, pages 106–113, 2001.

[VTF02] S. Vazhkudai, S. Tuecke, and I. T. Foster. Replica selection in the globus data grid. 2002.

[Wan07] J. Wang. The application of grey systems theory in hydropower business valuation. In *Proceedings of the IEEE International Conference on Grey Systems and Intelligent Services, 2007. GSIS 2007*, pages 504–508, Nov. 2007.

[Wen08] K. Wen. A matlab toolbox for grey clustering and fuzzy comprehensive evaluation. In *Proceedings of the Advnced Engineering Software*, volume 39, pages 137–145, Oxford, UK, Feb. 2008. Elsevier Science Ltd.

[Wes] Westgrid. www.westgrid.ca.

[WHR98] G. Williams, M. Hegland, and S. Roberts. A data mining tutorial. In *Proceedings of the International Conference on Parallel and Distributed Computing and Networks, PDCN'98*, Dec. 1998.

[WJ95] O. Wolfson and S. Jajodia. An algorithm for dynamic data allocation in distributed systems. volume 53, pages 113–119, Amsterdam, The Netherlands, The Netherlands, January 1995. Elsevier North-Holland, Inc.

[Wol97] R. Wolski. Forecasting network performance to support dynamic scheduling using the network weather service. In *Proceedings of the The Sixth IEEE International Symposium on High Performance Distributed Computing*, pages 316–325, 1997.

[WQZW09] H. Wang, J. Qi, W. Zheng, and M. Wang. Balance k-means algorithm. In *Proceedings of the International Conference on Computational Intelligence and Software Engineering, 2009. CiSE 2009*, pages 1–3, Dec. 2009.

[YLJ03] J. Yao, X. Li, and L. Jia. A new method based on ltb algorithm to mine frequent itemsets. In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics*, volume 1, pages 71–75, Nov. 2003.

[YLN06] D. Yamaguchi, G. Li, and M. Nagai. On the combination of rough set theory and grey theory based on grey lattice operations. In *Proceedings of the 5th International Conference on the, Rough Sets and Current Trends in Computing, RSCTC 2006, Kobe, Japan*, pages 507–516, Nov. 2006.

[YLN07] D. Yamaguchi, G. Li, and M. Nagai. A grey-based rough approximation model for interval data processing. In *Proceedings of the Information Sciences*, volume 177, pages 4727–4744, New York, NY, USA, Nov. 2007. Elsevier Science Inc.

[YYCW06] C-T. Yang, I-H. Yang, C-H. Chen, and S-Y. Wang. Implementation of a dynamic adjustment mechanism with efficient replica selection in data grid environments. In *Proceedings of the 2006 ACM symposium on Applied computing, SAC '06*, pages 797–804, New York, NY, USA, 2006. ACM.

[ZAZB00] E. W. Zegura, M.H. Ammar, F. Zongming, and S. Bhattacharjee. Application-layer anycasting: a server selection architecture and use in a replicated web service. In *Proceedings of the Transactions on Networking, IEEE/ACM*, volume 8, pages 455–466, Augest 2000.

[ZS03] N. Zsolt and V. Sunderam. Characterizing grids: Attributes, definitions, and formalisms. In *Proceedings of th Journal of Grid Computing*, volume 1, pages 9–23, March 2003.

[ZZLL09] D. Zhang, J. Zhang, K-K. Lai, and Y. Lu. An novel approach to supplier selection based on vague sets group decision. In *Proceedings of the Expert Systems Applications*, volume 36, pages 9557–9563, Tarrytown, NY, USA, July 2009. Inc. Pergamon Press.

Vita

Rafah M. Almuttairi is currently a research scholar in University of Hyderabad, India. She is faculty member at Babylon University, Iraq. She received her Master Degree in Computer Science from Baghdad University, Iraq in October, 2003. She has received her Bachelor Degree in Computer Science and Bachelor Degree in Physics Science from Babylon University, Iraq in October, 2001 and October 1994 respectively. She has worked as a Computer Trainer in USA's institute, National Democracy Institute (NDI). She has worked as a Database designer in USA's institute, Research Triangle Institute (RTI). She has authored several international conferences papers in area of grid computing. Her current research interest is in the area of data grid architecture and fuzzy decision making for grid resources and replica selection.