ANALYZING EFFECT OF DATA FRAGMENTATION ON DATA MINING ALGORITHMS IN DISTRIBUTED GRID ENVIRONMENT

By

Abdul Latif

(2002-NUST-BIT-102)

A project report submitted in partial fulfillment of
the requirement for the degree of
Bachelors in Information Technology

In

NUST Institute of Information Technology (NIIT)
National University of Sciences and Technology (NUST)
Rawalpindi, Pakistan
(2007)

CERTIFICATE

Certified that the contents and form of project report entitled "**Analyzing Effect of Data Fragmentation on Data Mining Algorithms in Distributed Grid Environment"** submitted by Abdul Latif have been found satisfactory for the requirement of the degree.

Advisor: _____

Professor Dr. Arshad Ali

Co-Advisor:_____

Mr. Shahrzad Khattak

Member: _____

Dr. Ashiq Anjum

Member: _____

Dr. Aamir Shafi

DEDICATION

I would like to dedicate my humble work to my family, my teachers, and friends and to NIIT where I spent a tough but meaningful part of my life, praying for your success.

ACKNOWLEGMENT

I will thank first of all my CREATOR my Lord Allah for giving me power and courage to complete my final year project successfully. I don't have words to thank my family, my father, mother and elder brother, who always encouraged me in the hardest and despair situations during my degree and have supported me financially beside all the hurdles. They were always there to pray for me all the time when I really needed it.

I am extremely thankful to Professor Dr.Arshad Ali, being my adviser he supported me through the course of my project and provided me every means through which I can complete my degree project. The one person I won't be able to forget whole of my life is Dr.Ashiq Anjum. Being my committee member he was always like a mentor to me. He always helped me in my research career and was always pushing me to complete my work and do the best job in order to succeed. I would like to thank Mr.Shahrzad Khattak and Dr.Aamir Shafi who beside their busy schedule always gave me time to resolve my issues and problems.

I will also thank all my friends especially Izhaar Ul Hassan, Irfan Habib and Kamran Soomro for making my graduation easy for me. I would like to say special thanks to my seniors Faisal Khan, Muhammad Adeel Zafar, Muhammad Atif, Tahir Azim and Waqas-ur-Rehman for guiding me in my project in one way or the other. Working with them provided me a good opportunity to learn from them and get an experience, which will always help me in my practical life.

TABLE OF CONTENTS

ABSTRACT

Grid data mining has emerged as an important field, as data continues to be produced at astounding rates and, in order to get most out of this data, efficient techniques for data mining are required. Various classes of algorithms have been developed. Because of the increasingly large magnitude of the data to be processed the Grid has become a natural platform for Data mining. Recently a lot of frameworks have been developed which facilitate grid data mining. However Grids, being very application specific, vary in terms of data distribution and scale. The performance of different classes of data mining algorithms against varying levels of data distribution has not been studied. This project aims to cover this domain, in that it tries to benchmark data mining algorithms on Grids in order to determine if some class of algorithms are more suitable for some specific level of data distribution in Grids. This work will facilitate the deployment of optimized data mining algorithms on application specific Grids and may lead to a generic adaptive Grid data mining framework in future.

The simulations are run on Sun Fire V890 system. The java bases data mining platform Weka is being used along with the GridSim, grid simulation environment. On the basis of this analysis we will be able to have a relation between number of computation nodes, data fragmentation level and the performance in terms of time. Thus on the basis of this analysis we can design a new data mining platform for distributed Grid infrastructure where the data can be efficiently and intelligently distributed to the grid resources in order to minimize the whole data mining time.

List of Figures

List of Tables

Chapter 1

INTRODUCTION

In recent years, the grid-based computing paradigm has attracted much attention. The most important, main and beneficial aspect of Grid computing is the sharing of distributed computing resources (such as software, hardware, data, sensors, etc) [1]. The two main types of Grid computing i.e. computational grids and data grids aim to focus on handling compute intensive jobs and data intensive jobs respectively. Grid-based computing is being used in several application areas including biomedicines, physics, astronomy, ecology and various fields of sciences. In health sciences and biomedicines [2] researchers are building infrastructures of networked high-performance computers, data integration standards, and other emerging technologies, to pave the way for medical researchers to transform the way diseases are being analyzed, treated and controlled. Several efforts [2] are being made to develop such infrastructures; sophisticated data mining algorithms and efficient schemes for communication which can integrate distributed gigantic data sources and are able to extract knowledge from these data sources.

Extracting knowledge from distributed, heterogeneous and gigantic data sources is not a simple task. It is quite challenging to extract knowledge from these distributed data sources. The point that how we can make use of the distribution of data is very crucial, currently the data distribution is a big hurdle in the data mining process of these data sources. What if we make use of the data distribution to minimize the total time taken by mining the integrated data of the concerned

data sources? On the other we can have a scenario where the data available on a certain machine is very large as compared to the computational power available at that machine in order to mine the data at that location in optimistic minimal time. If the data is independent in nature then the data at that site can be divided into several chunks and then sending to appropriate computational resources we can not only achieve the aim of mining that data but also can minimize the total time data mining process by applying the same data mining algorithm on each chunk of the data in parallel.

This work aims on the concept that by simulating various data mining algorithms on the Grid infrastructure with different level of data fragmentation, on the basis of the analysis of the results generated we can find a relation between the number of computational resources available (computation nodes), the level of data fragmentation and the amount of time it will take to mine the data. Thus we can be able to decide how to achieve the data mining of a gigantic data in minimal time by dividing the data in to several chunks.

1.1 GRID COMPUTING

Grid Computing has emerged as new area of research and a potential technology to serve the high requirements of data storage and computation in the future. It is analogous to an electric power grid where power generating sources are located distributed but the user is able to use and access electric power without bothering about its location in an easy and simpler way. The grid technology paradigm aims in similar way to provide a ubiquitous source of computing power for users throughout the world, just like the electrical power. It provides a

mechanism of establishing VO (virtual organizations). It enables sharing the heterogeneous computing and storage resources of organizations and individuals which are distributed across the globe to form a massive computing environment (i.e. VO) through which complex and large-scale problems can be solved. But once there is resource sharing in distributed environment then huge security problems arises and grid technology aims to prevent unauthentic access and abuse of these resources. Thus the user is provided with an illusion that a virtual super computer is present which has a tremendous computation as well as storage power and that can serve ones requirements for data storage as well computation. Grids consist of networks of computers, storage, and other devices that can pool and share resources. But these resources are provided to only authorized and authenticated users in order to prevent the security breaches. Grid computing architecture can be defined in several ways as several people have their own definitions, but in simpler words Grid computing can be defined as a system that has:

- Standards that make it possible for heterogeneous systems and applications to share compute and storage resources transparently
- Proper network architecture to connect the distributed resources.
- And a schema for intelligent load-balancing over a high speed network

In an article in 2002 [3] that motivate the need for Grid computing, Ian Foster describes the Grid "vision" as:

"...to put in place a new international scientific infrastructure with tools that, together, can meet the challenging demands of 21st-century science."

If we see today then Ian Foster rightly said, because there are numerous research projects working on different aspects of grid computing including development of the core technologies, deployment and application of grid technology to different scientific domains.

1.2 GRID COMPUTING USAGE

Grid computing is about getting computers to work together and utilize their idle processing and storage power. The main goal of grid computing is the maximum utilization of resources. It is observed that almost every organization is having enormous and unused computing capacity but many of them are not being used up to their maximum capacity. It is observed that most PCs do nothing for 95% of a typical day like a common user personal computer; Mainframes remain idle 40% of the time, UNIX servers (a huge collection over the web) are actually "serving" something less than 10% of the time so a huge amount of resources are being wasted. How to use these available resources in a constructive manner? The answer is Grid computing where you can utilize these computation and storage resources up to maximum level in a fully authenticated and secure environment.

Data production in different fields of science is increasing day by day. One of fields of science producing large amount of data is biomedical data. Also other research and routine activities like in CERN CMS, are generating huge amount of data regarding physics experiments. This data is useless until it is not mined for some useful information. Thus digging out hidden patterns out of this data is really helpful to scientists to come up with a result for the experiments and analysis they are doing. The explosive computing environment formed by Grid has proven to be

so significant that scientists working to solve many of the difficult scientific problems have realized the potential of such shared distributed computing system and have started to utilize theses systems to solve their problems relating to data mining engineering designs and high energy physics such as analysis of data generated CMS at European Organization for Nuclear Research. The wide distribution of the data over the globe makes the data mining process on these data sources a good candidate to run on the distributed Grid infrastructure.

1.3 **PROBLEM STATEMENT**

Analyze the effect of Data Distribution/Fragmentation on the data mining algorithms in the distributed Grid infrastructure in order to minimize the total time of the data mining process.

1.4 **MOTIVATION**

As the journey of science proceeds to advancement it is being guided by the analysis of data. Taking an example, huge genome sequences [4] available online motivates collaborative research in biology, catalogs of sky surveys [5] [6] enables astronomers to answer queries that may have taken years of observation, high-resolution, long-duration simulation data from experiments and models enables research in physics, ecology, geological sciences, chemistry, biomedicines and many more fields. Once Ian Foster said that converting data into scientific discoveries requires *"connecting data with people and computers"*. And in this process one has to:

- Find the data of interest.

- Move the data to desired locations.
- Manage large scale computations
- Schedule resources on data and
- Manage the proper security, authorization and authentication.

Taking an instance CERN, the main goal of CERN, the European Organization for Nuclear Research in Geneva, Switzerland is to study the fundamental structure of matter and the interaction of forces. In particular, subatomic particles are accelerated to nearly the speed of light and then collided. Such collisions are called events and are measured at time intervals of only 25 nanoseconds in four different particle detectors of the Large Hadron Collider (LHC) CERN's next generation accelerator which has started data collection in 2006. According to the MONARC Project1 each of the 4 main experiments will produce around 1 Petabyte of data a year over a life span of about two decades. This data needs to be analyzed by about 5,000 physicists around the world. Since CERN experiments are collaborations of over a thousand physicists from many different universities and institutes, the experiments' data is not only stored locally at CERN but is distributed worldwide in so called Regional Centers (RCs), in national institutes and universities called Tier 1 centers.

The aim to analyze the data in distributed environment motivates the need for *Grid* environments. To extract meaningful information from distributed, heterogeneous data repositories on the grid, sophisticated knowledge discovery architectures have to be designed. The process of data mining on the grid is still a relatively new area of research. While several architectures have been developed for this purpose, the framework of *distributed* data mining on the grid

infrastructure still has a long way to go. The purpose of this research oriented project is analyze the effect of data fragmentation on the data mining algorithms in the distributed grid environment and making the data mining process of grid infrastructure efficient and fast.

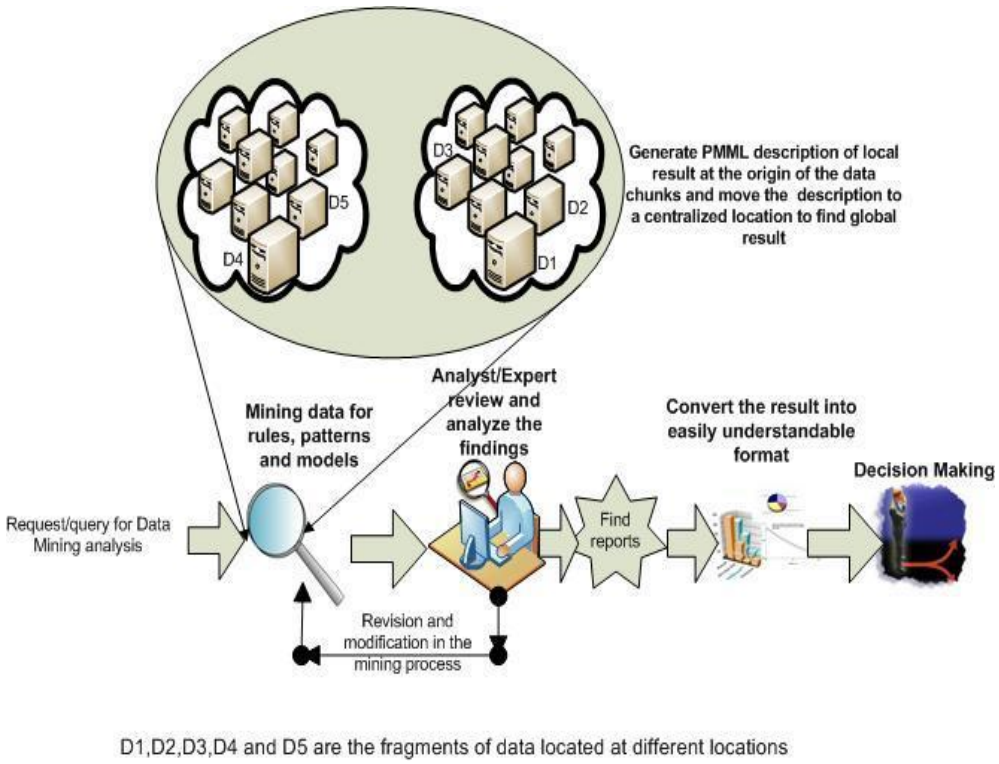

Figure 1: An approach to Knowledge Discovery on Grid Infrastructure

1.5 AIM AND OBJECTIVES

Data mining has been determined to be a compute and memory intensive application [1]. Because the magnitude of the data is so large [2], the Grid is a natural platform for mining these kinds of data. However in a Grid there are challenges in terms of communication latencies between Grid nodes, in terms of fragmentation/replication of the data, heterogeneity in terms hardware and software and distribution of the data. I have tried to explore the question that are

some data mining algorithms better suited to certain types of Grids. This project aims to study if the performance of data mining algorithms is correlated with the level of data fragmentation in a Grid environment and if generic adaptive algorithms be developed to effectively address the challenges posed by Grids. This project also addresses the issue of how effective existing Data mining algorithms tackle data distribution in Grids.

Chapter 2

LITERATURE REVIEW

2.1 RELATED WORK

Recently some efforts have been made towards developing Grid centric data mining solutions, most notable efforts include, GridMiner[7], DataMiningGrid[8], KnowledgeGrid[9], Grid Weka[10] and Discovery Net[11] and Weka4WS[12]. These frameworks allow for Grid centric data mining.

2.1.1 GridMiner

The knowledge discovery process in distributed grid environment is a very challenging task. The data integration of heterogeneous distributed data sources and then its analysis poses great risks and issues those needed to be tackle. GridMiner is one of the first initiatives to serve the purpose of data mining on distributed grid environment. Advanced medical application of Traumatic Brain injury was used to test the validity of the developed system. Some the very early requirements GridMiner considered were to tackle the distributed data as medical data related to Traumatic Brain Injury is distributed all over the globe

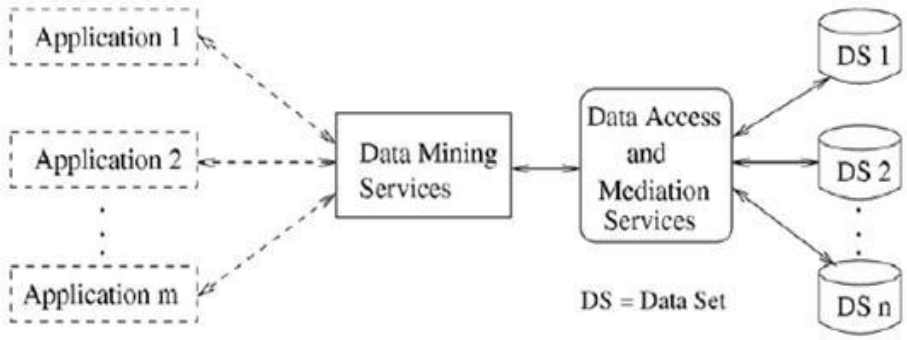

Figure 2: Data Access, Mediation and Data Mining Services

Reference: GridMiner: An Infrastructure for Data Mining on Computational Grids Peter Brezany, Jürgen Hofer, A Min Tjoa, Alexander Wöhrer

GridMiner can work in both types of data distribution scenarios i.e. data available on a single machine and data available on multiple machines.

2.1.2 DataMiningGrid

DataMiningGrid, one of the several efforts towards distributed data mining on the Grid Infrastructure is built over the Globus Tool Kit and using various open source technologies. The system has been tested to be useful for various fields of scientific research and businesses in the quest of knowledge discovery. The main challenge undertaken by DataMiningGrid was to develop a generic data analysis and knowledge discovery tool which will work for all kind of data.

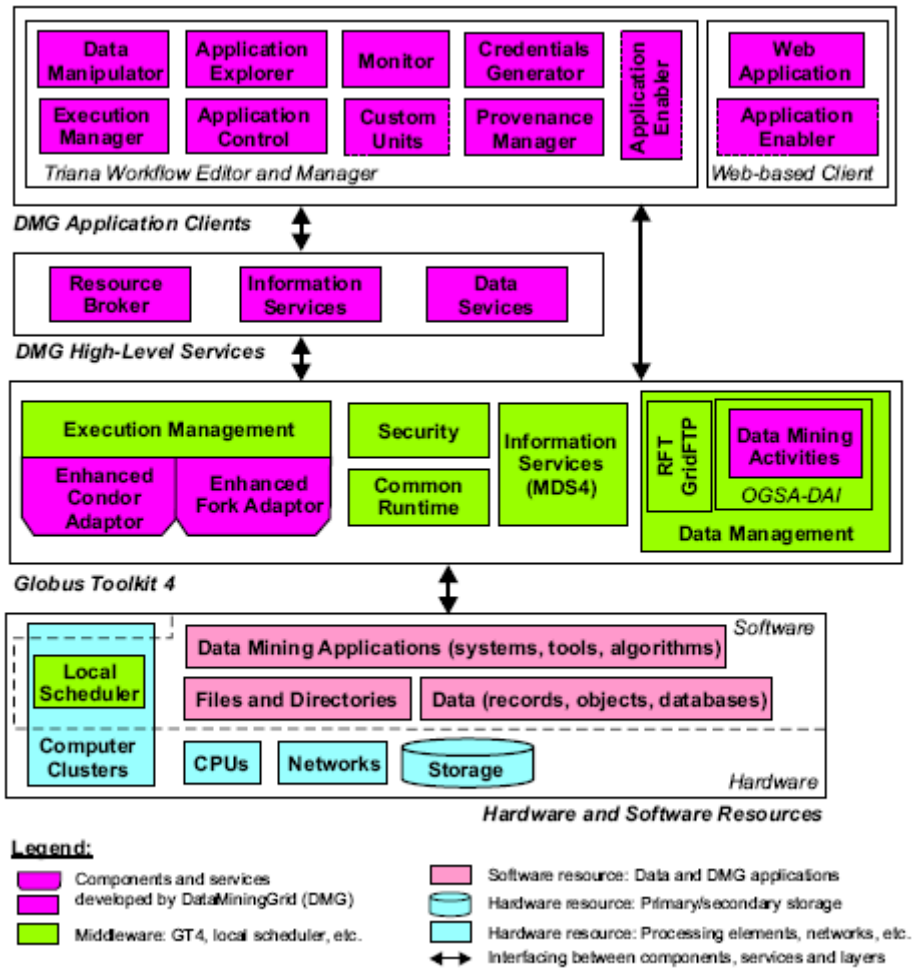

Figure 3: DataMiningGrid Architecture

Reference: Grid-enabling data mining applications with DataMiningGrid: An architectural perspective, Vlado Stankovski, Martin Swain etal.

The architectural diagram of DataMiningGrid shows several layers. In general the higher layers make use of the lower layers. The Grid Resource Layer provides the hardware and software resources required for the execution of the whole environment. The grid middleware layer is responsible for providing the core grid middleware services to the DataMiningGrid System. The higher level layer consist of Grid Resource Broker, data service and information service. The information service is very crucial to monitor the characteristics and status of the resources and jobs. The resource broker is responsible for scheduling the resources

to the jobs and the data services provide a convenient way to access and manipulate the data sources on the DataMiningGrid.

2.1.3 KnowledgeGrid

The data distribution across the globe and the computational resources required to mine the data in efficient manner compels the scientists to use some software solutions in the form of distributed or parallel data mining. Grid technology paradigm has emerged as a strong candidate these days to serve the resource requirements of distributed data mining. The project Knowledge Grid is is a grid based distributed data mining software and its architecture is built upon the computational Grid. It utilizes the common services of grid infrastructure and by using the extra services for knowledge discovery exploits the resources of grid technology where each node ultimately serve as a sequential or parallel computer providing either data storage services or computational services.

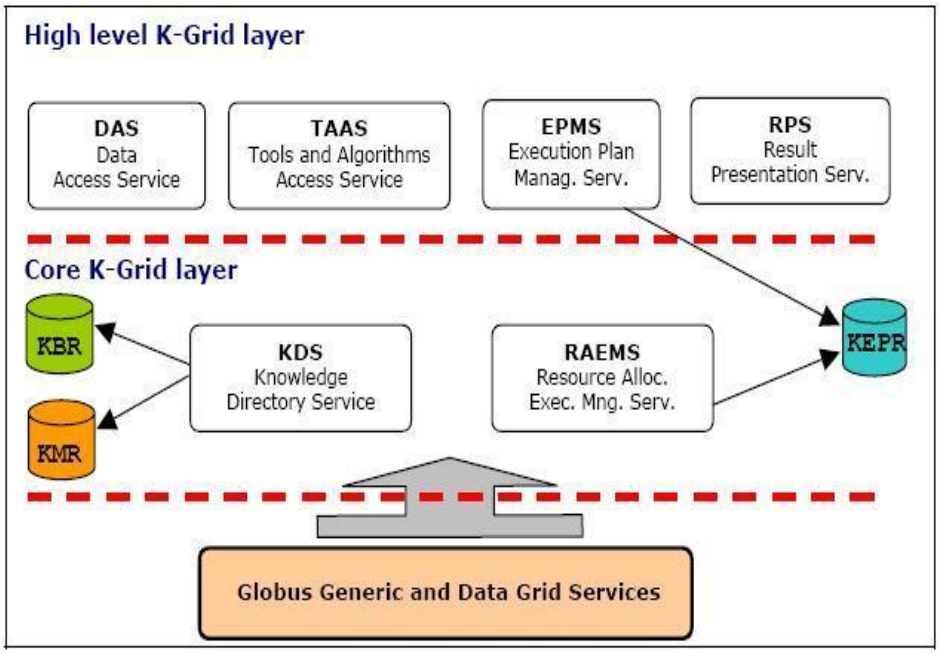

Figure 4: Knowledge Grid Architecture

Reference: KNOWLEDGE GRID, An Architecture for Distributed Knowledge Discovery , Mario Cannataro and Domenico Talia

Knowledge Grid aims to address the following issues

- Handling gigantic data
- Difference in the schemas of the integrating data sources or simply data heterogeneity.
- Security and data privacy, scalability and openness.
- Also the main issue is its compatibility with the grid infrastructure and grid awareness.

The core and common services of grid infrastructure are Grid Monitoring Service, scheduler, steering service, estimator service etc. In knowledge Grid the services implemented are basically of two main types i.e. core knowledge Grid services and high level knowledge grid services used to perform the data mining in distributed environment. The core Knowledge Grid

Services provides the whole execution plan for the data mining over Grid infrastructure and the high level knowledge grid services actually executes the plan of data mining.

2.1.4 Discovery Net

Discovery Net aims to provide a generic knowledge discovery solution but its approach mainly originated from the needs of information extraction in the field of bioinformatics. In bioinformatics usually complex data pipelined approach is used where models are built and those models are used to be applied on the coming data in order to get the data mined and this process continues. Being built on the data resource federation layer it classifies knowledge discovery services into two categories i.e. Computation services and Data services. The computation service is in simple words a data mining algorithm which actually provides the intelligence and logic of mining the data. The computation service can have constraints like location and platform. The main purpose of data service is to provide metadata for analysis and to make a data set from the distributed resources.

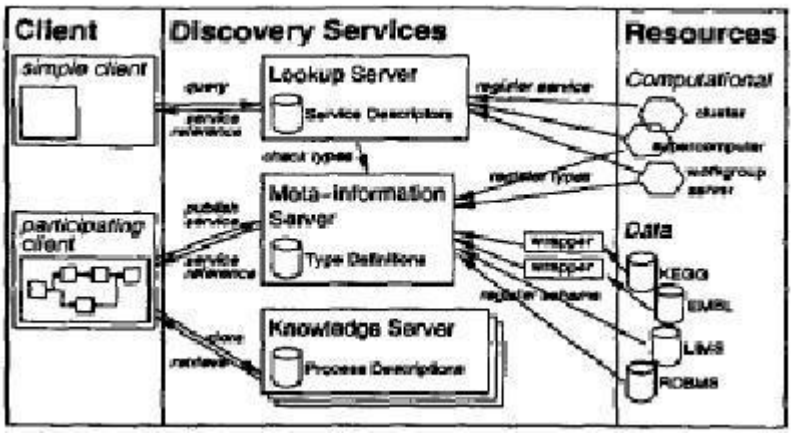

Figure 5: DiscoveryNet Architecture

Reference: Discovery Net: Towards a Grid of
Knowledge Discovery V. CurSin M. Ghanem, Y. Guo M. Kohler, A. Rowe J. Syed, P. Wendel

A client is used by the user to construct and define discovery procedures and process. In DiscoveryNet the Resource Discovery Server is a knowledge base of published service definitions and performs the role of resource resolution server, as resources are never requested directly from a client but only through a request to resolve the location of a particular service irrespective of the location.

2.1.5 Weka4WS

Weka4WS, is the extension of the famous Weka data mining and machine learning toolkit for distributed Grid data mining. a framework that extends the Weka toolkit for supporting distributed data mining on Grid environments. Weka4ws is built upon the Globus Toolkit 4. Instead of adopting the Open Grid Services Architecture (OGSA) it adopts the emerging Web Services Resource Framework (WSRF) for accessing remote data mining algorithms and managing distributed computations over the grid infrastructure.

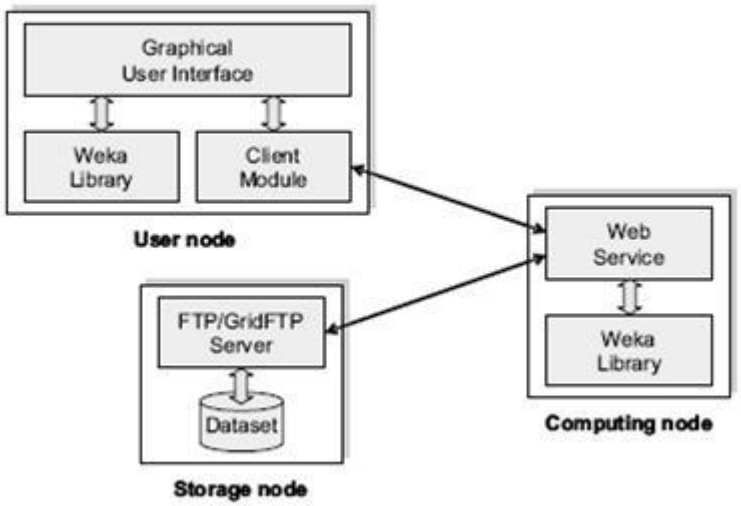

Figure 6: A general Architecture of Weka4WS Framework

Reference: Weka4WS: a WSRF-enabled Weka Toolkit for Distributed Data Mining on Grids.

<http://www.datamininggrid.org/wdat/works/att/ljudoc002.content.05904.pdf>

Weka4ws supports both local and global data mining. It characterizes the nodes into three types i.e. User Nodes, Computation Nodes and Storage Nodes. The User Nodes are the machines where the user is resided and initiate requests. The computation nodes are used to process the actual data mining process and the storage nodes contain the data to be mined.

2.2 CLASSICATION OF DATA MINING ALGORITHMS

Data mining algorithms are known to be specific on the type and nature of the data. While analyzing the data and its true nature, usually it can be categorized into four types of data forms i.e. textual, temporal, transactional and relational data forms [13]. The data mining techniques and algorithms can be divided into six categories on the basis of the knowledge aimed to be extracted from the target

data, i.e. Association Rules Mining, Data Classification, Clustering Analysis, Trends Discovery, Patterns Discovery and Data Description or Summarization [13]. In association rules mining we try to find association between different items from the data. The famous algorithms are Apriori, DIC, Partition and Eclat. Association rule mining is usually applied on the transactional data forms but in some cases when the data is in advance form such as in "group by transaction" , then it can also be applied on the relational data forms. Data classification aims at putting data into predefined classes on the basis of similarity with the classes. Decision trees, neural nets, Bayesian classifiers, Support Vector Machines and cased-based learning are few of the data classification techniques. Classification algorithms are most suitable for relational data forms. Clustering analysis groups the data into natural grouping without any predefined classes. Some well known algorithms are K-Means, EM etc. These are most suited for relational data forms. The trends analysis approach aims to find the trends in data values as they change with time and are generally used in making predictions. The most suitable data types for this type of analysis are temporal data forms. Pattern recognition techniques search for patterns in data similar to already existing or known patterns, the famous algorithms are Linear Classifier, Quadratic Classifier and K Nearest Neighbour Classifier. Pattern recognition techniques are normally applied on textual data forms and temporal data forms. The last category is that of the data description or summarization techniques. They observe the data stored in data bases with higher views which are used to represent rules that can explain concepts and help humans in understanding the nature of the data. The most suitable data types for this type of analysis are relational data forms.

The above mentioned approaches and data forms are standard and general in nature. These techniques are being used in centralized data mining but variations of the algorithms for distributed and parallel architectures are also available and research work is being carried out to improve their efficiency. Which approach is better and which is not depends upon the requirements of the analyst. The efficiency and accuracy of the results of different algorithms is also application specific and cannot be predetermined. However, general trends can be identified, which is what we plan to do in this paper. The results presented here can help analysts in the selection of a particular algorithm according to their needs.

2.3 GRIDIFICATION OF ALGORITHMS

Section 2.2 briefly discusses different methods and approaches of data mining and also the criteria for the selection of algorithms and techniques for one's own purpose. Data mining applications and data mining tasks are highly computation intensive. When the data to be analyzed is in huge amounts i.e. in terabytes and petabytes then the processing or computational time it takes to extract the knowledge from these massive data sources is significant and it needs to be minimized. There are many infrastructures supporting the parallelization of tasks like clusters, but Grid technology has emerged as an efficient platform in recent years to deal with such computation intensive and data intensive jobs like data mining Grid technology provides an efficient solution for issues like load balancing, security, fault tolerance, memory management etc by exposing heterogeneous computational and data storage resources to the user. The requirement for the gridification of data mining algorithms lies in the fact that, in large geographically distributed organizations and institutions the data is stored in

various data warehouses, owned by different organizations and is in different formats located in different parts of the globe. In order to analyze this distributed heterogeneous data, an underlying network and infrastructure is needed, for which Grid Technology is suited perfectly. Using this global infrastructure we can analyze the distributed data using specially designed data mining algorithms for the Grid infrastructure or extension of the algorithms developed for distributed-memory parallel architectures. The gridification of algorithms depends upon the nature of the target data, if the data to be analyzed is dependent then the approach to divide the data and applying the algorithm on each part of the data is not feasible. Because in the case of dependent data communication cost between each chunk (algorithm working on each chunk separately) will be significant and will produce a lot of delay in the final result. For this reason, it is preferable that the data to be mined remain at the point of origin and only the results [14] be moved instead of moving the actual data. In case of independent data which we are using for our simulation, the above stated approach can work efficiently.

2.3.1 Issues in Gridification of Data mining Applications

Enabling the existing data mining algorithms to run on Grid infrastructure, or designing new ones to cope with the distributed heterogeneous data located all over the globe is very challenging. These challenges include integration of distributed heterogeneous data, data warehousing problem over the Grid infrastructure, data ownership issues, security issues etc. The most crucial one among these challenges is that of the data integration issues of heterogeneous data like biomedical data. The data is distributed and if there is the requirement to mine

the combined data of more than one data sources, we need a combined schema so that the data can be analyzed without missing any information [15]. Secondly there is no concept of data warehouses on the Grid infrastructure. The reason behind this is that we usually deal with large gigantic data sources in the distributed Grid environment and combining multiple data sources to be analyzed requires a lot of complexities like resolving heterogeneity of semantics, data types and schemas of the involved data sources, transferring huge amount of data over the limited bandwidth networks, security and data ownership aspects of the data being transferred and used. These are some of the important issues related to the gridification of data mining algorithms and application. Regarding the data distribution and fragmentation aspects of data mining over the Grid infrastructure we discuss the two possible scenarios of data distribution and data mining algorithms flow.

2.3.1.1 Centralized data sources

If the data is located at a single location then analyzing this kind of data is quite simple. One has to apply the required data mining algorithm on the data looking into the nature and characteristics of the target data and get back the result. However the complexity comes in when the computational resources available at the origin of the data are not quite sufficient to process the data in the desired time span. Then we need to move the data to some computational resource or resources based on the requirements. For this purpose the data needs to be divided into chunks, if the data is independent then it's easy but in case of dependent data, division into chunks is quite tricky and needs in depth understanding of the data

and its nature. Furthermore the security of the data and the bandwidth limitations of the network must also be considered while moving the data.

2.3.1.2 Decentralized data sources

If the data is located at more than one location then we have the following two types of scenarios:

1 Each location or site contains the same amount of information or attributes of the data. We can also term it as horizontally partitioned data. e.g. all the branches of Standard Chartered Bank keep the same amount of information about their clients. Although the size of the data can vary. There can be two ways to mine this data, either analyze the data at its origin if there are some restrictions due to security, ethical and data ownership issues, or mine the data at suitable computational resources where the data can be moved accordingly.

2 The second kind of data distribution is termed as vertical portioning. Each site or location has different information and attributes of the data. In order to have global consistent data unique identifiers are used to relate the data in the distributed environment. There is high dependency between the data located at different locations. Normally the data distribution is column wise, every site have a subset of the original data columns.

2.4 WHERE AND WHEN CAN WE DO THE GRIDIFICATION?

So far we have seen that how much Grid data mining can be useful as compared to centralized data mining while dealing with distributed data sources. Now, in order to decide where to do gridification of algorithms and where not, we will have to understand the nature of the data available either it is located in one location i.e. centralized or decentralized, are we authorized and able to move the data from its origin or not? Is there any security and data ownership issues? Are the computational resources available can compute the task in due time? If we are moving data from various nodes to a central location, is there enough storage available at that site? Is the distributed data dependent or not? Once these types of questions and many more like these are answered then one can decide well wither to do gridifcation or not. In order to clarify the usage of gridification of data mining tasks, the following scenarios are discussed:

As discussed earlier if the data is located in one location then we can simply utilize the available computational resources if they are able to process the mining task in due time else we have to move the data to some suitable resource.

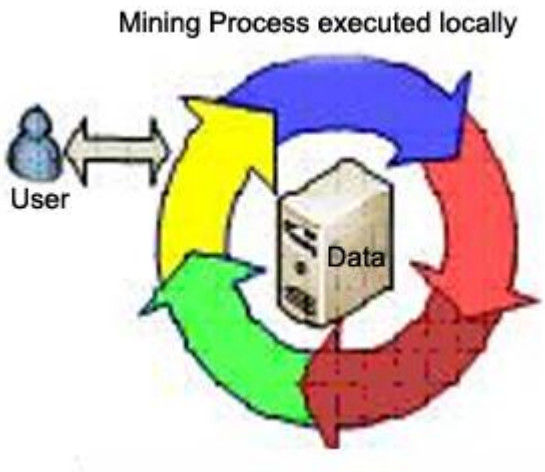

Figure 7: Data Mining Process executed locally on the available computational resources at the origin of the data

In case of decentralized distributed data, we have two options; either move the parts of data from various locations to one central node and then integrate the data and apply the data mining algorithm. This is only possible if we have no security or data ownership restrictions on the target data.

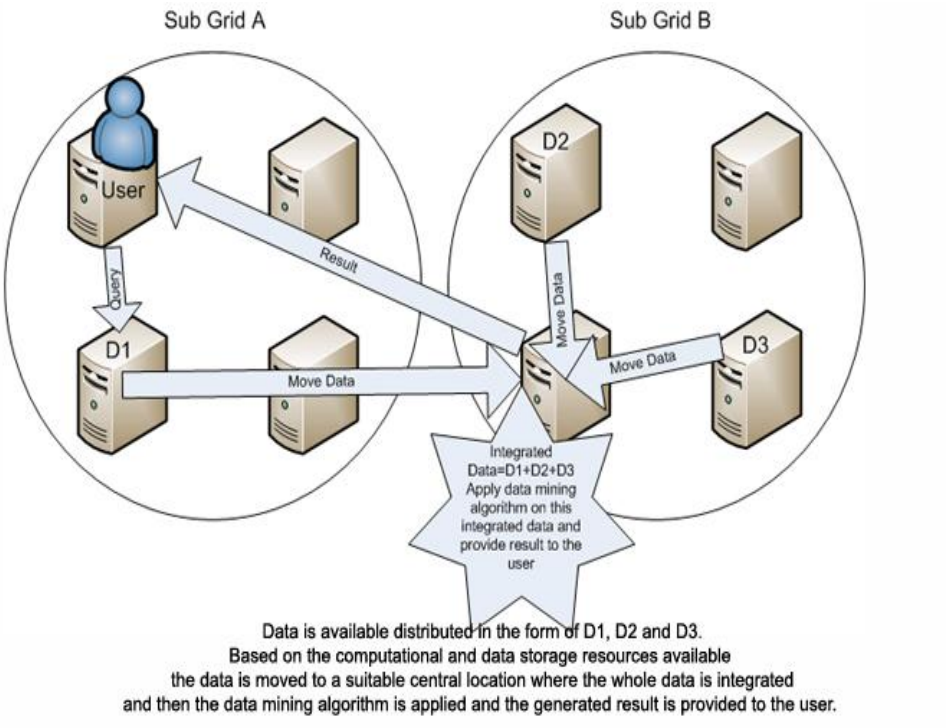

Figure 8: Grid data mining: Data Mining at optimal suitable location.

Let suppose, we are not allowed to move the data from its origin, now we have the only option to mine the data at that site. The local results are generated at each data chunk's side and then moved to a central location to be integrated as a global result.

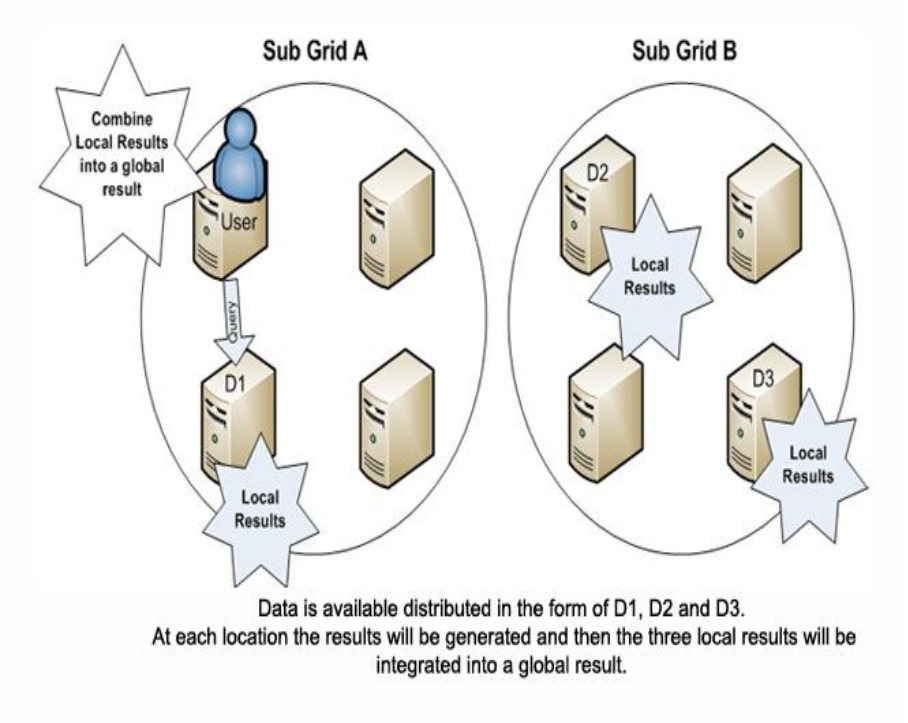

Figure 9: Grid data mining: Data Mining at the origin of the data.

Thus only and only if we have the exact idea of the data location and nature we can decide what type of data mining can be useful and efficient in the given scenario. Off course the size of data also matters, because moving a gigantic data source over a low bandwidth, low speed network will ultimately choke the network.

Chapter 3

SIMULATION AND RESEARCH METHODOLOGY

3.1 TEST BED

Simulations were conducted to study the effect of fragmentation on the performance of data mining algorithms. For this purpose, an event-based simulation tool called GridSim 4.0 was used.

To perform the actual data mining, we used gridified versions of the centralized, cluster-based data mining algorithms. In order to gridify them we used a distributed data mining model. In this model, each site performs mining on its local data store. Once the local mining is complete, a description of the mined data is generated using an interface language called PMML. The central grid server uses the description generated by each site to generate a global picture of the data. In this way centralized, cluster-based data mining algorithms are used for grid data mining. The process is depicted in figure 10.

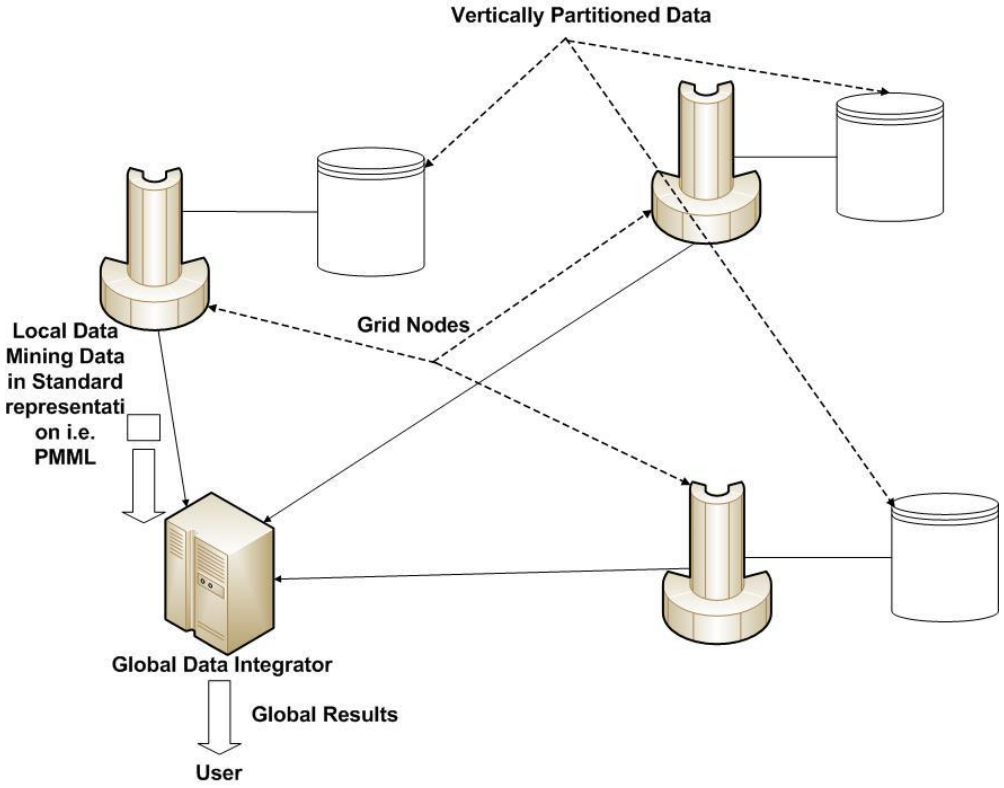

Figure 10: Grid Data Mining- A Model based approach

The data was distributed across the grid in a vertical fashion. Each data store within a particular site contained a portion of the data. This allowed us to evaluate the performance of the algorithms on fragmented data.

3.2 SUMULATION EVIRONMENT AND PARAMETERS

Following is a description of the simulation environment:

We simulated a grid of 100, 150 and 200 sites. Since the data was already distributed across the grid, and each instance of the data mining algorithm was run locally at each site without any dependencies, the network connectivity amongst the sites didn't matter. The data was fragmented across the grid. The actual number of

sites containing the data depends on the fragmentation level. A 10% fragmentation level means that 10% of the nodes actually contain the data. Initially 5GB data was used for simulation. But due to the small size of data the results had a very little difference.

Three types of data was used i.e. CPU, weather and contact-lenses data. This data was generated by a python script. The data was randomly generated by generating data of each attribute between its highest and lowest value.

Data was generated randomly using Python scripts (Appendix A). A total dataset of 500 MB was generated, which was distributed across the grid with above levels of fragmentation.

3.3 GRIDSIM: GRID MODELLING AND SIMULATION TOOLKIT

GridSim toolkit provides a comprehensive facility for simulation of different classes of heterogeneous resources, users, applications, resource brokers, and schedulers and is very efficient in simulating grid scheduling algorithms prior to actual deployment. It can be used to simulate application schedulers for single or multiple administrative domains distributed computing systems such as clusters and Grids, it is the choice of the user to provide number of resources. Application schedulers in the Grid environment, called resource brokers, perform resource discovery, selection, and aggregation of a diverse set of distributed resources for an individual user. Each user has his or her own private resource broker and hence it can

be targeted to optimize for the requirements and objectives of its owner. In contrast, schedulers, managing resources such as clusters in a single administrative domain, have complete control over the policy used for allocation of resources. This means that all users need to submit their jobs to the *central* scheduler, which can be targeted to perform global optimization such as higher system utilization and overall user satisfaction depending on resource allocation policy or optimize for high priority users.

The key features of GridSim are.

- It allows modeling of heterogeneous types of resources.
- Resources can be modeled operating under space- or time-shared mode.
- Resource capability can be defined (in the form of MIPS (Million Instructions Per Second) as per SPEC (Standard Performance Evaluation Corporation) benchmark).
- Resources can be located in any time zone.
- Weekends and holidays can be mapped depending on resource's local time to model non-Grid (local) workload.
- Resources can be booked for advance reservation.
- Applications with different parallel application models can be simulated.
- Application tasks can be heterogeneous and they can be CPU or I/O intensive.
- Multiple user entities can submit tasks for execution simultaneously in the same resource, which may be time-shared or space-shared. This feature

helps in building schedulers that can use different market-driven economic models for selecting services competitively.

- Network speed between resources can be specified.
- It supports simulation of both static and dynamic schedulers.
- Statistics of all or selected operations can be recorded and they can be analyzed using GridSim statistics analysis methods.

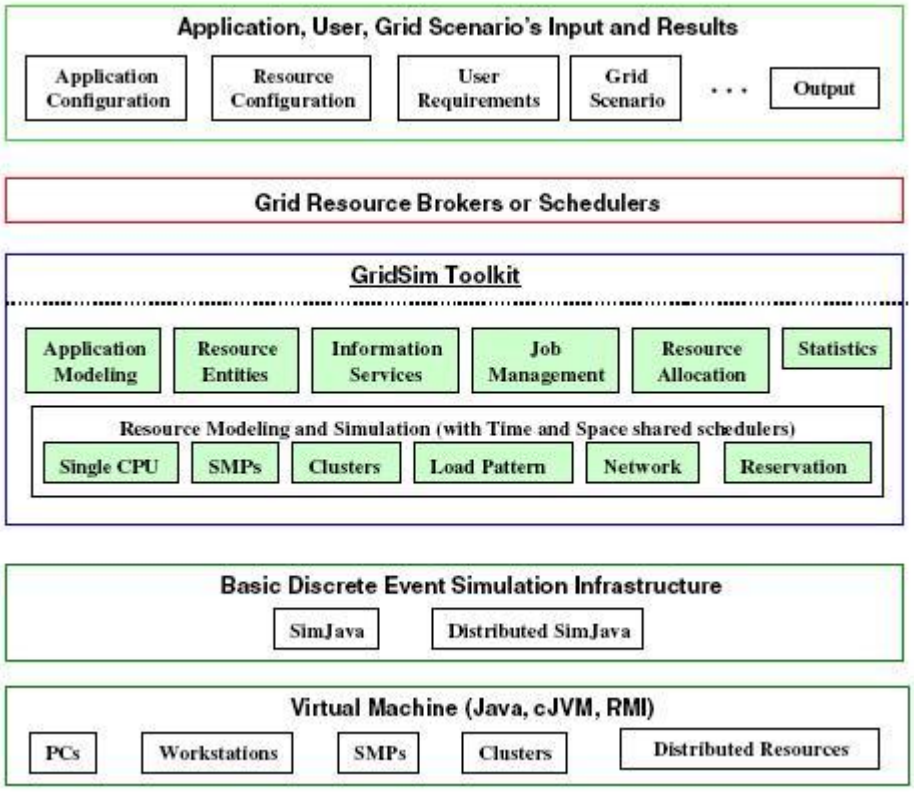

Figure 11: A modular architecture for GridSim flatform and components
Reference: GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing (2002) Manzur Murshed, Rajkumar Buyya, David Abramson

Chapter 4

ALGORITHMS USED FOR SIMULATION

4.1 ALGORITHMS USED FOR SIMULATION

Analysis of data can result in patterns, rules or groupings. The ultimate result of data mining task is the information which makes the basis for decision making, predicting future situations, detection of relations, explicit modeling, clustering, classification, association rules mining, deviation detection and for a lot more purposes.

The classification of data objects to various groupings without the availability of any prior classes is unsupervised classification or clustering. It is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. We use a number of clustering algorithms namely Cobweb, expectation maximization, farthest first, and k-means for experimentation purposes on our distributed fragmented data. The clustering COBWEB creates, is expressed in the form of a tree, with leaves representing each instance in the tree, the root node representing the entire dataset, and branches representing all the clusters and sub-clusters within the tree. An expectation-maximization (EM) algorithm is used in statistics for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent

variables. It maximizes the likelihood of the clustering being formed when randomly drawing from the data set. In Farthest First algorithm a random point is selected as the center, and then the k most distant points are computed from it. K-means creates random centers, then moves them until all instances are clustered and distances are minimized.

Classification algorithms are used to classify data into groups using quantitative information of the data. It is used as a tool to map information from a dataset X to a dataset Y. We used a number of classification algorithms on our set of data to classify it, such as NaiveBayes, KStar, and Decision Tables. The NaiveBayes classifies a dataset 'X' into a specific class 'C' based on the probability of the features of X i.e. f1, f2 ... fn, belonging to the class 'C'. KStar is an Instance-based learner using an entropic distance measure. Instance-based learners classify an instance by comparing it to a database of pre-classified examples. A decision table is a scheme that produces rules formatted as a table, from selected attributes.

Association rules mining shows the relationship of two data objects in a transaction and the effect of one data object on another object. The association rules are patterns discovered in data that includes the concept of transaction, basket or group. Association rules relate the items within each transaction or basket, as it is usually related to transactional data and is termed as Market Basket Analysis also.

Above are some of the algorithms which will be used for our simulation and experimentation purposes. But initially we are using only J48 classifier, Naïve Bayes and Bayes Net algorithms.

Chapter 5

TESTING AND RESULTS

5.1 ASSUMPTIONS

For the simulation purposes we have the following assumptions:

The data to be analyzed is independent in nature and thus there is no issue in dividing the data into chunks and then analyzing each chunk separately.

- The data can be moved from the point of its origin and there is no restriction on the data movement to be analyzed.
- The data is already available at the computing node in the specified chunks, therefore we are not dealing with the cost of moving the data from its origin to the computing node.
- The network bandwidth is not taking in consideration because in GridSim we explicitly fix it.
- The results produced for this small size of data will correspond to large gigantic size data and the behavior of the algorithms will be the same in general.
- We are visualizing to use the Model based approach to the data mining i.e. at each local chunk the Local Model will be generated and then it all the Local Models will be moved to a central location where they would be aggregated to have a Global Data Mining Model.

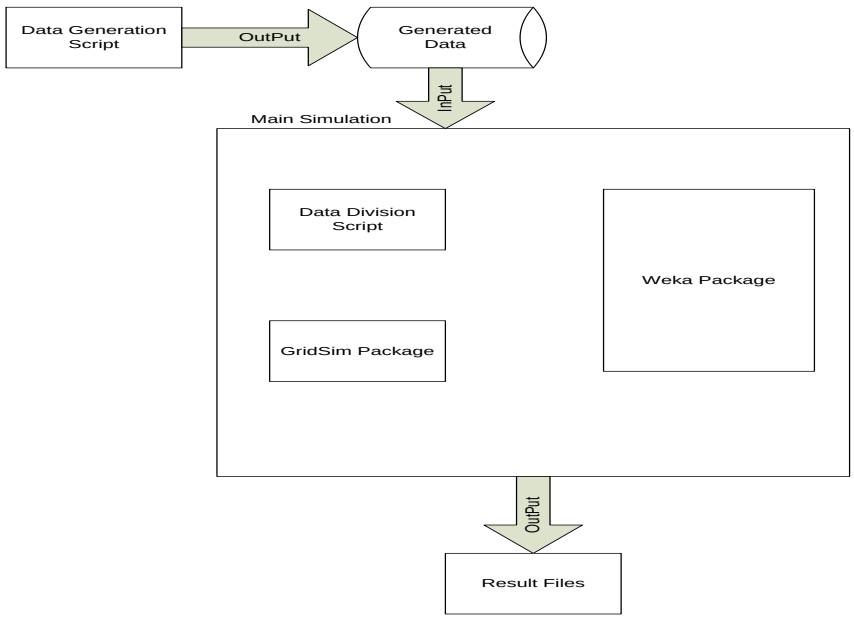

Figure 12: Simulation Set Up

5.2 SIMULATIONS

We have analyzed two main techniques of data mining that is clustering and classification.

The problems related to classification are one area where machine learning algorithms have been found to be particularly useful and are widely used. Classification can be defined in simple words as an act of grouping or distributing things on the basis of their resemblance provided that the classes must be known in prior. Application domains for classification algorithms include speech and handwriting recognition, internet search engine, optical character recognition, credit score analysis, biomedical field and other scientific areas like physics and astronomy. Till now significant number of classification algorithms has been proposed, the performance and efficiency of the algorithms vary from one application to another.

And hence, we try to analyze which classification algorithm is best suited for various fragmentation levels of the data.

We study here two classification algorithms and compare their performance on various level of data fragmentations. The algorithms are:

- J48 Algorithm, and implementation of C.45 algorithm
- NaiveBayes Algorithm, based on Bayesian Theorem.
- BayesNet Algorithm

J48 Classification Algorithm:

For easy interpretation decision trees are popular knowledge representations. The learned patterns using J48 are represented as a tree and nodes in the tree embody decisions based on the values of attributes and the leaves of the tree provide predictions starting from the root. This algorithm is Weka's implementation of the C4.5 decision tree learning algorithm and was invented by Ross Quinlan.

Naïve Bayes Algorithm:

This is a statistical learning algorithm and it applies a simplified version of Bayes rules for a test scenario on the data given the tested data. This method is simple and fast learning and work efficiently in various scenarios.

Simulation Results

Data: contact-lenses.arff

Data Size: 500MB

Simulation Environment: Weka and GridSim packaged on SUN Fire V890 System.

Algorithm: weka.classifiers.trees.J48

For result data please refer to Appendix C.

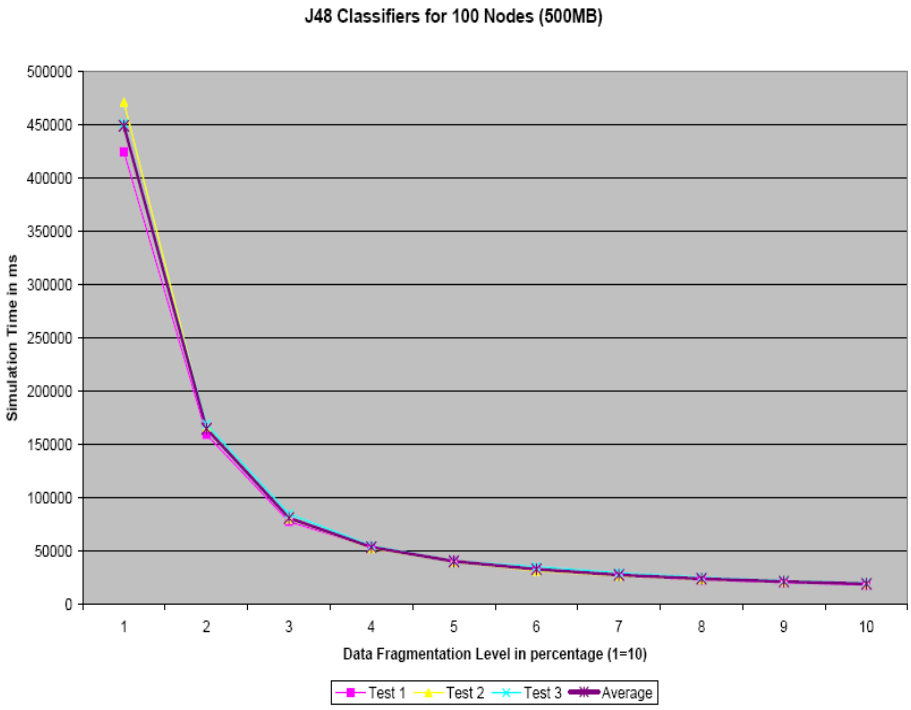

Figure 13: J48 Algorithm for 100 Nodes

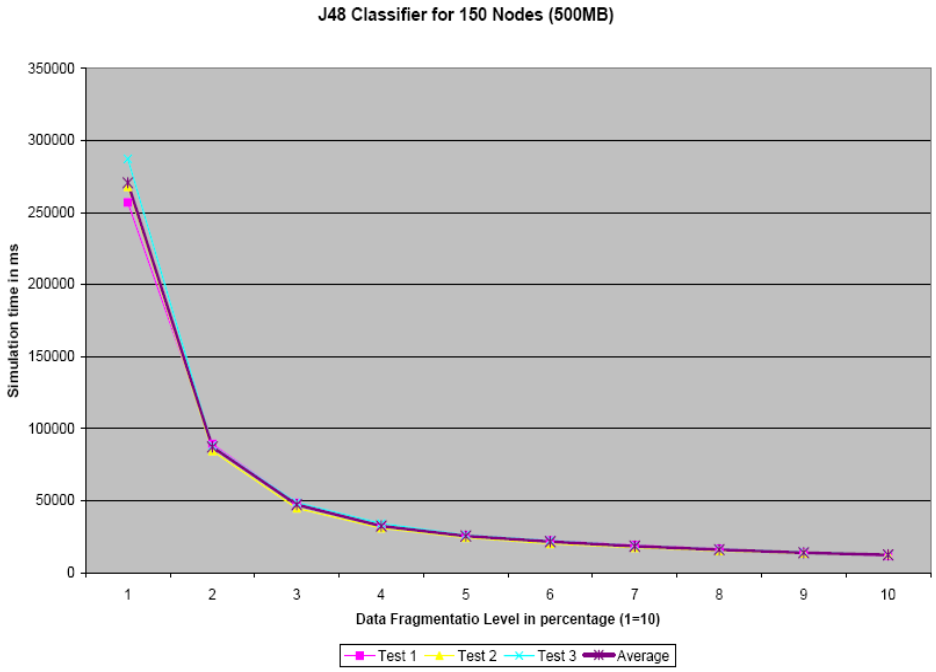

Figure 14: J48 Algorithm for 150 Nodes

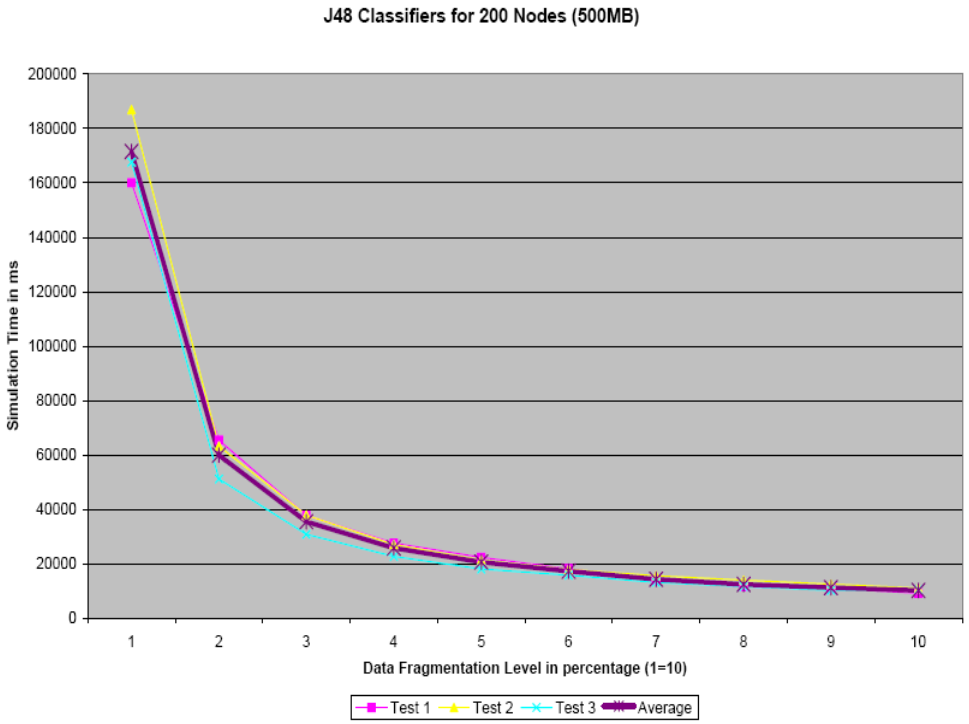

Figure 15: J48 Algorithm for 200 Nodes

Data: contact-lenses.arff

Data Size: 9.8MB

Simulation Environment: Weka and GridSim packaged on SUN Fire V890 System.

Algorithm: weka.classifiers.bayes.BayesNet

For result data please refer to Appendix D.

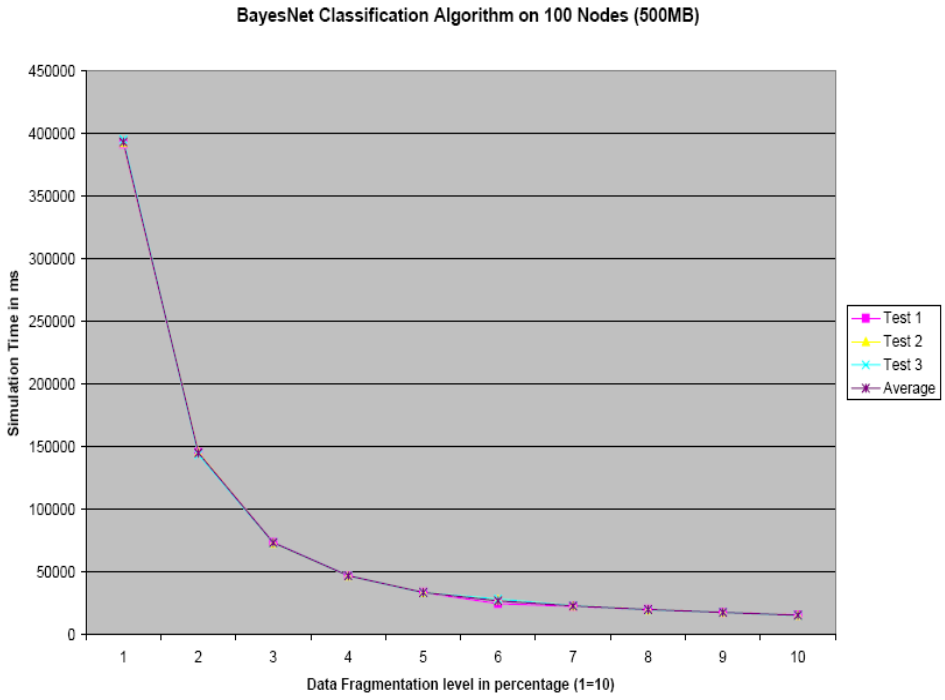

Figure 16: BayesNet Algorithm for 100 Nodes

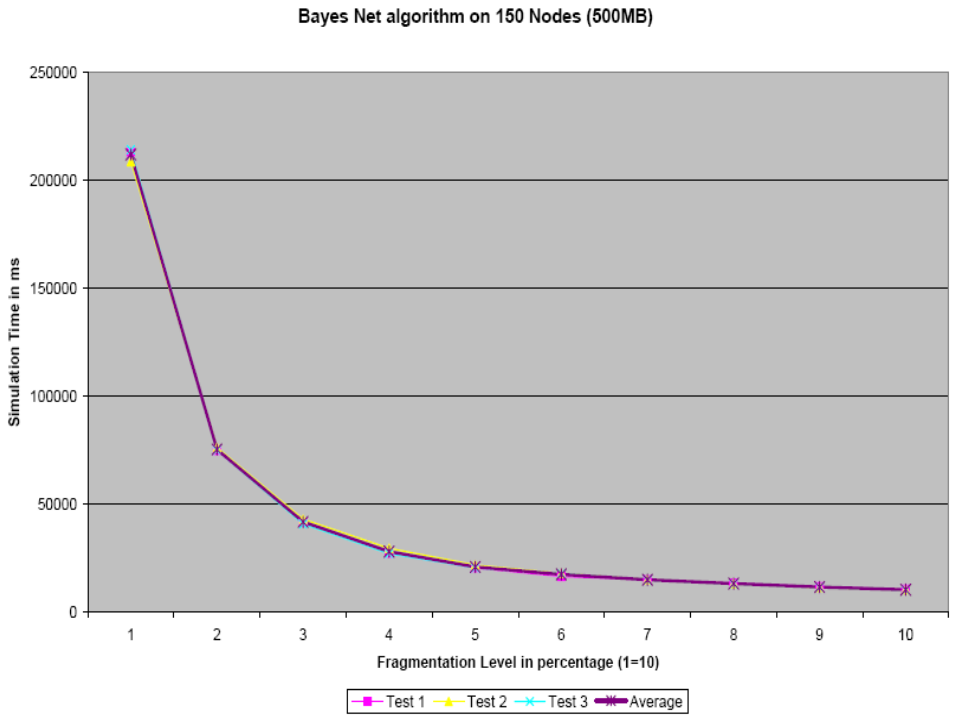

Figure 17: BayesNet Algorithm for 150 Nodes

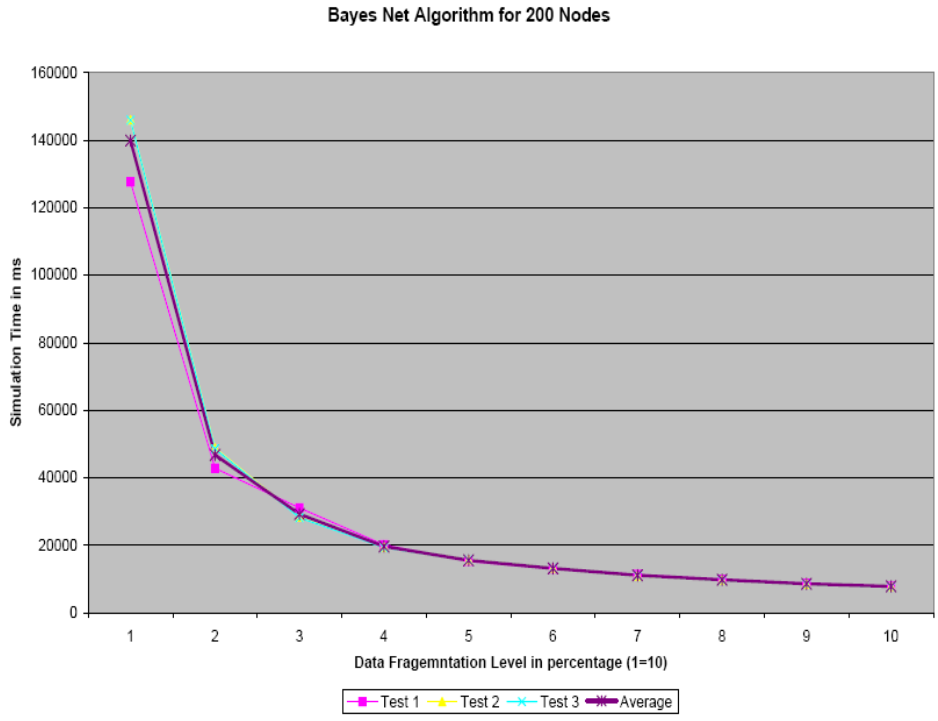

Figure 18: BayesNet Algorithm for 200 Nodes

The above six figures i.e. Figure 13 to Figure 18 shows the application of J48 and NetBayes classification algorithms on 100,150 and 200 available nodes respectively and for varying level of data distribution. There is a clear tendency that as we add the computation resources and also decrease the data size by increasing the fragmentation level, the execution time decreases and thus it rightly supports the principles of Grid computing and parallel computing. One can see a very big drop of execution time between the 10 % data fragmentation and the 20 % data fragmentation, the reason is that the whole data is halved and due to the small data size this drop is significant. If we have a large data set then this drop will be as normal as the rest of the drops in execution time.

Further the results can be more refined by executing the experiments several times keeping the data constant.

5.3 COMPARISON OF J48 AND BayesNet CLASSIFICATION ALGORITHMS

Here we have compared the two classification Algorithms and the results are shown in the following graphs.

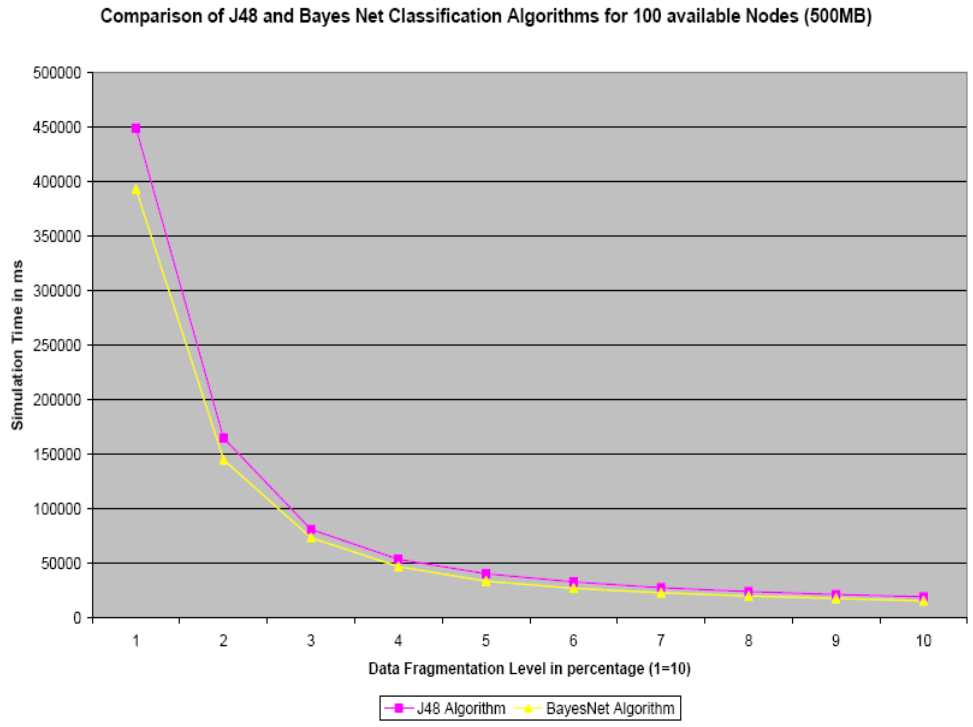

Figure 19: Comparison of Classification Algorithms for 100 Nodes

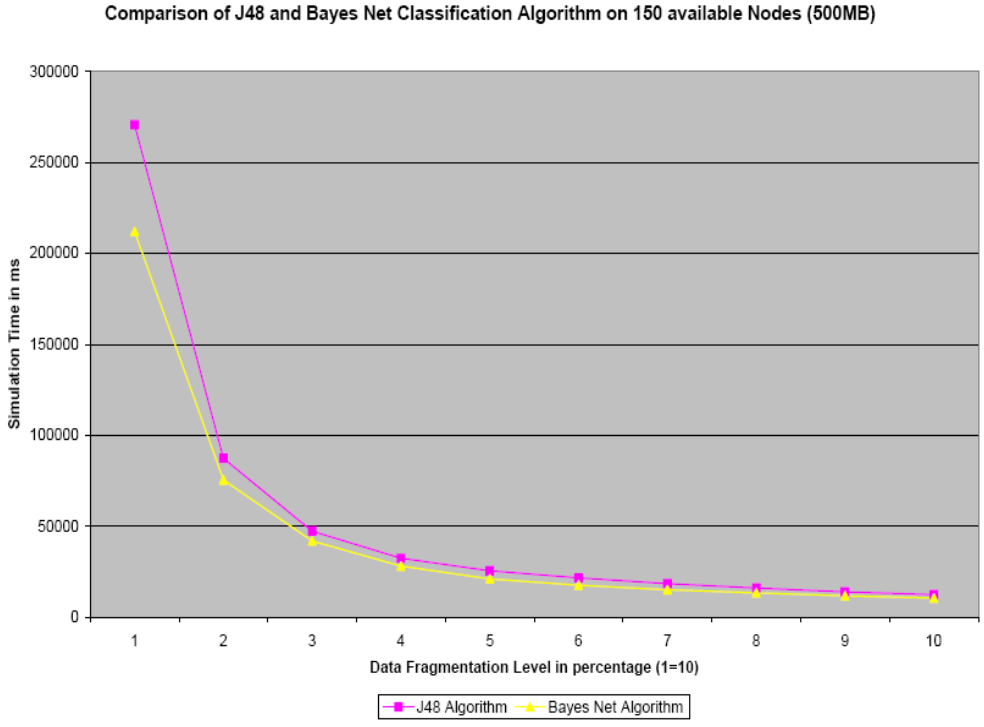

Figure 20: Comparison of Classification Algorithms for 150 Nodes

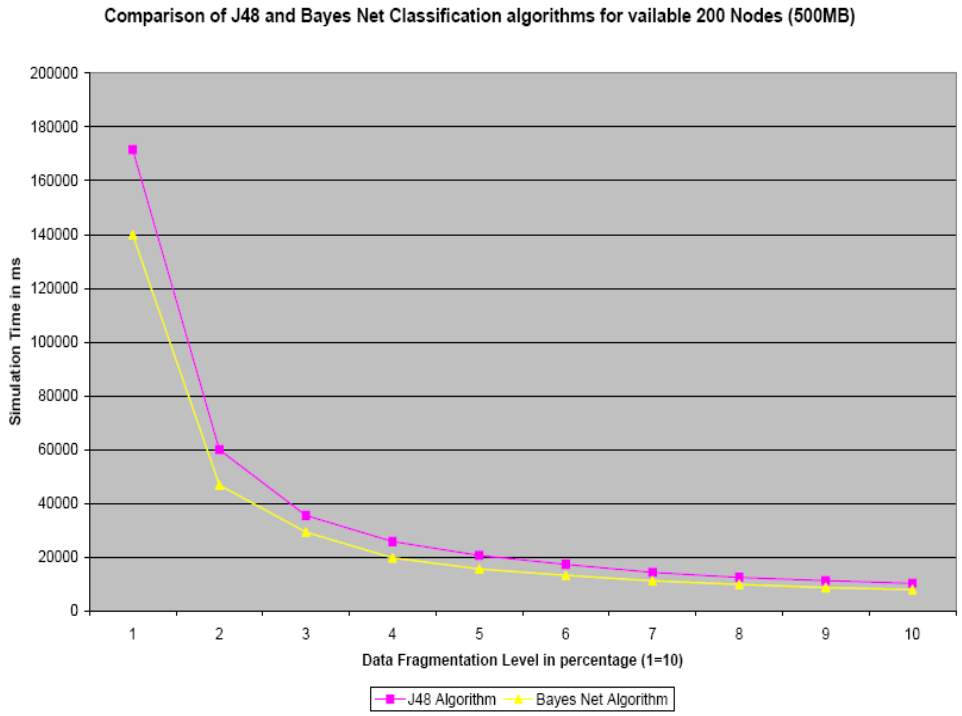

Figure 21: Comparison of Classification Algorithms for 200 Nodes

The comparison of J48 and Bayes Net Classifiers shows that Bayes Net Algorithm is more efficient than J48 algorithm and the difference will be more notable on a large data size.

Chapter 6

CONCLUSION AND FUTURE DIRECTIONS

Looking into the results the outcome can be summarized as:

"The distributed nature of the data sources in recent times can be exploited to minimize the over all data mining time of the whole integrated data by using the Grid infrastructure"

Thus we can exploit the distributed data to minimize the data mining time provided that the computing resources are added in the distributed grid infrastructure. The existing algorithms still can work better to achieve the goal of time minimization of the data mining by having a good framework and criteria to distribute the data over the computing and storage resources. Although I had the assumptions that the data is in equal size of chunks it might be the case that it is in arbitrary size. There might be the case that the data available on a certain node might not be able to be analyzed by the local computing power so one has to divide the data and to move it to the candidate computing resources where it can be mined quickly, in this process the data division criteria and the data movement must be tackle which in this work was my assumptions.

If I would have more time I must analyze the algorithms for dependent data and then will implement the system in real life so as to have some business application.

ACHIEVEMENTS

A research paper is being written on this project “Effects of data fragmentation on data Mining Algorithms in Grid Infrastructure”. Only results are pending.

REFERENCE

1. Haimonti Dutta, "Empowering Scientific Discovery by Distributed Data Mining On a Grid Infrastructure", a Proposal for Doctoral Research, 2006.
2. "Health-e-Child", March 16th, 2007, < http://www.health-e-child.org/about>
3. Ian Foster, "The Grid: A new infrastructure for 21st century science." Physics Today, Feb 2002.
4. "The human genome project", April 20th, 2007, <http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml.>
5. "Sloan Digital Sky Survey", April 25th, 2007, <http://www.sdss.org.>
6. 2-Micron All Sky Survey. May 27th, <http://pegasus.phast.umass.edu.>
7. Brezany, P., Janciak, I., and Tjoa, A. M. "GridMiner: A Fundamental Infrastructure for Building Intelligent Grid Systems", IEEE/WIC/ACM international Conference on Web intelligence (WI'05). - Volume 00 (September 19 - 22, 2005). WI. IEEE Computer Society, Washington, DC
8. "DataMiningGrid" , June 20th, 2007, <http://www.datamininggrid.org>
9. Cannataro, M. and Talia, D, "The knowledge grid", Commun. ACM 46, 1 (Jan. 2003), 89-93
10. "Grid Weka", June 15th, 2007. <http://smi.ucd.ie/~rinat/weka/>
11. Ćurčin, V., Ghanem, M., Guo, Y., Köhler, M., Rowe, A., Syed, J., and Wendel, P. 2002. "Discovery net: towards a grid of knowledge discovery", KDD '02. ACM Press, New York, NY, 658-663.

12. “Weka4WS: a WSRF-enabled Weka Toolkit for Distributed Data Mining on Grids”, February 10th, 2007. <http://www.datamininggrid.org/wdat/works/att/ljudoc002.content.05904.pdf>
13. Mon-Fong Jiang Shian-Shyong Tseng, Shan-Yi Liao, “Data Types Generalization for Data Mining Algorithm”, 1999, IEEE.
14. Matthias Klusch, Stefano Lodi and Gianluca Moro “Agent-Based Distributed Data Mining. The KDEC Scheme”, 2003.
15. Ghoting, A., Buehrer, G., Parthasarathy, S., Kim, D., Nguyen, A., Chen, Y., and Dubey, P. 2005. “A characterization of data mining algorithms on a modern processor”. DAMON '05. ACM Press, New York, NY.
16. Patrick Wendel, Moustafa Ghanem, “Scalable clustering on the data grid”, All Hands Meeting, 2005.

Appendices

Appendix A

Data Generation Script for CPU Data (takes arguments, outputfile and size of the data)

```
#!/usr/bin/env python


import os
import random
import sys


def usage_error():
	print "Usage: ", sys.argv[0], "output-file size[K/M/G]"
	sys.exit(1)
if len(sys.argv)!=3:
	usage_error()
size_attributes_values={'K':'000', 'M':'000000', 'G':'000000000'}
size_attribute=sys.argv[2][-1]
maxsize=int(sys.argv[2][0:-1]+size_attributes_values[size_attribute])


attributes_max=[1000, 10000, 100000, 200, 10, 200, 1000]
```

```
filename=sys.argv[1]
output=open(filename, 'w')

attributes_string="\n".join(["@relation 'cpu'", "@attribute MYCT real", "@attribute MMIN real", "@attribute MMAX real", "@attribute CACH real", "@attribute CHMIN real", "@attribute CHMAX real", "@attribute class real", "@data"])

output.write(attributes_string+"\n")

while os.stat(filename)[6]<maxsize:
    data = []
    for n in range(500):
        temp=[]
        for i in attributes_max:
            temp.append(str(random.randint(0, i)))
        data.append(",".join(temp)+"\n")
    output.writelines(data)
output.close()
```

Appendix B

Sample CPU Data

```
@relation 'cpu'
@attribute MYCT real
@attribute MMIN real
@attribute MMAX real
@attribute CACH real
@attribute CHMIN real
@attribute CHMAX real
@attribute class real
@data
125,256,6000,256,16,128,199
29,8000,32000,32,8,32,253
29,8000,32000,32,8,32,253
29,8000,32000,32,8,32,253
29,8000,16000,32,8,16,132
26,8000,32000,64,8,32,290
23,16000,32000,64,16,32,381
23,16000,32000,64,16,32,381
23,16000,64000,64,16,32,749
23,32000,64000,128,32,64,1238
```

Appendix C

Simulation Results-J48-contact-lenses

Data: *contact-lenses.arff*
Data Size: 500MB
Simulation Environment: Weka and GridSim packaged on SUN Fire V890 System.

Algorithm: weka.classifiers.trees.J48

Experiment

Table 1: Experiment -J48 Algorithm results

S.No	No of Compu-tation Nodes	Data Fragme-ntation	Simulation Time Test 1 (ms)	Simulation Time Test 2 (ms)	Simulation Time Test 3 (ms)	Average Simulation Time (ms)	Average Simulation Time (mints)
1	100	10	424936.4	471667.3	451484.4	449362.7	7.489378
2	100	20	159559.05	166441.55	169158.25	165053	2.750883
3	100	30	77890.9	80637.83	85210.2	81246.31	1.354105
4	100	40	53768.9	52934.25	55434.17	54045.77	0.900763
5	100	50	40864.26	40107.3	40955.1	40642.22	0.67737
6	100	60	32178.23	31579.96	35695.68	33151.29	0.552522
7	100	70	27189.32	26768.22	29904.87	27954.14	0.465902
8	100	80	23207.65	23802.6	25786.17	24265.47	0.404425
9	100	90	20694.13	21651.61	22531.94	21625.89	0.360432
10	100	100	18482.49	19643.83	20349.34	19491.89	0.324865
11	150	10	256935.46	267887.93	287308.46	270710.6	4.511844
12	150	20	89342.8	84598.33	88130.03	87357.05	1.455951
13	150	30	48347.95	44786.57	48666.73	47267.08	0.787785
14	150	40	32134.3	31010.31	34173.01	32439.21	0.540653
15	150	50	25999.68	24362.86	26220.88	25527.81	0.425463
16	150	60	22308.42	20156.6	22363.17	21609.4	0.360157
17	150	70	19106.52	17432.07	18757.49	18432.03	0.3072
18	150	80	16556.05	15182.5	16401.45	16046.67	0.267444
19	150	90	13848.14	13527.45	14297.61	13891.07	0.231518
20	150	100	11854.1	12914.63	12405.25	12391.33	0.206522
21	200	10	160069.45	186934.7	167714.45	171572.9	2.859548
22	200	20	65404.2	63200.07	51128.1	59910.79	0.998513
23	200	30	37724.63	37834.76	30824.8	35461.4	0.591023

24	200	40	27584.7	27054.33	22665.32	25768.12	0.429469
25	200	50	22313.87	21353.52	18053.52	20573.64	0.342894
26	200	60	18092.34	17876.06	15764.19	17244.2	0.287403
27	200	70	14071.33	15583.44	13130.52	14261.76	0.237696
28	200	80	11618.88	13961.41	11608.53	12396.27	0.206605
29	200	90	10989.88	12483.65	10176.46	11216.66	0.186944
30	200	100	9159.03	11016.69	10432.27	10202.66	0.170044

Appendix D

Simulation Results-BayesNet

Data: contact-lenses.*arff*

Data Size: 500 MB

Simulation Environment: Weka and GridSim packaged on SUN Fire V890 System.

Algorithm: weka.classifiers.bayes.BayesNet

Experiment

Table 2: BayesNet Experiment Result

S.No	No of Compu-tation Nodes	Data Fragme-ntation	Simulation Time Test 1 (ms)	Simulation Time Test 2 (ms)	Simulation Time Test 3 (ms)	Average Simulation Time (ms)	Average Simulation Time (mints)
1	100	10	391997.2	393088.6	395450.4	393512.1	6.558534
2	100	20	145737.6	145808.5	144100.	145215.4	2.420256
3	100	30	73894.96	73427.6	73559.63	73627.4	1.227123
4	100	40	47357.82	47332.12	47249.32	47313.09	0.788551
5	100	50	33917.54	33920.64	33848.18	33895.45	0.564924
6	100	60	25087.45	28325.06	28675.25	27362.59	0.456043
7	100	70	22963.17	23321.21	23375.3	23219.89	0.386998
8	100	80	20383.8	20265.61	20223.51	20290.97	0.338183
9	100	90	18111.75	17994.67	17970.46	18025.63	0.300427
10	100	100	15868.42	15718.79	15662.03	15749.75	0.262496
11	150	10	213216.86	208706.13	214349.26	212090.8	3.53484583
12	150	20	74955.53	76335.4	75069.06	75453.33	1.2575555
13	150	30	41674.46	43102.35	40951.88	41909.56	0.69849272
14	150	40	27470.3	29679.4	27185.78	28111.83	0.46853044
15	150	50	20583.08	21911.89	20571.17	21022.05	0.35036744
16	150	60	16615.95	18043.44	17975.38	17544.92	0.29241539
17	150	70	15080.81	15075.84	14993.2	15049.95	0.2508325

18	150	80	13546.83	13283	13240.40	13356.74	0.22261239
19	150	90	11897.38	11573.74	11614.88	11695.33	0.19492222
20	150	100	10628.22	10502.12	10396.21	10508.85	0.1751475
21	200	10	127693.1	146080.9	145984.5	139919.5	2.33199167
22	200	20	42843.52	48890.15	48611.65	46781.77	0.77969622
23	200	30	31087.7	28449.01	28193.65	29243.45	0.48739089
24	200	40	20136.12	19541.62	19369.05	19682.26	0.32803772
25	200	50	15288.38	15705.44	15645.21	15546.34	0.25910572
26	200	60	13045.66	13265.35	13237.05	13182.69	0.21971144
27	200	70	11366.66	10975.50	11047.62	11129.93	0.18549878
28	200	80	9927.09	9751.5	9701.4	9793.33	0.16322217
29	200	90	8732.02	8541.50	8559.67	8611.063	0.14351772
30	200	100	7934.02	7815.30	7741.90	7830.407	0.13050678