분산된 데이터마이닝을 위한 프레임워크의 설계 및 구현

Design and Implementation of a Distributed Data Mining Framework

Prakash Kadel and Ho-Jin Choi
School of Engineering, Information and Communications University
119, Munji-ro, Yuseong-gu, Deajeon, 305-714, Korea
{prakash, hjchoi}@icu.ac.kr

Abstract

We envisage that grid computing environments allow us to implement distributed data mining services, that is, those applications which analyze large sets of geographically distributed databases and information using the computational power and resources of a grid environment. This paper describes an experimental framework towards such a distributed data mining approach, including design considerations and a prototype implementation. Based on the "Knowledge Grid" architecture suggested by Cannataro et al., we identify four major components – user node, broker node, data node, and computation node – and define their individual roles. For implementing the prototype, we have investigated methods for utilizing distributed resources within a grid computing environment, e.g., communication and coordination among the various resources available.

1. Introduction

Data mining is a process that calls for huge computational and storage capabilities. These constraints often make the process of data mining time consuming and difficult to carry out. Utilizing the large no. of computational and storage resources available in a distributed environment such as grid can make the process more efficient and feasible.

In this paper we summarize the prototype of a data mining system. It is a simple system which be applied to the grid computing environment. The basic idea behind the system is based on The Knowledge Grid [1]. To utilize the distributed resources in a network we establish communication and coordinate among the various resources available on a network.

We have also tried to utilize the benefits that P2P mode of communication can provide to the distributed data mining. Here we describe the process of producing a classifier algorithm out of some dataset available in on the network.

Using P2P form of communication in data mining gives us a number of advantages. Most important of all the system becomes more decentralized and flexible. In this system, we have tried to make an extensive use of peer to peer communication.

The rest of the paper is organized as follows. Section 2 describes background and related works, section 3 describes the system requirements and components. Section 4 describes the system implementation details.

2. Background and Related Work

There has been a lot of work in this field. A lot of algorithm and systems have appeared. This system implementation of ours is based on the The Knowledge Grid. ADaM is another system that deals with disturbed data mining.

There has been a lot of work in this field. A lot of algorithm and systems have appeared. This system implementation of ours is based on the The Knowledge Grid. The architecture is built on top of a computational grid and provides access to high-end computational resources. It defines a set of additional layers on top of the basic grid services to implement the services of distributed knowledge discovery in a geographically distributed group of nodes (computing

and data resources).

Algorithm Development and Mining System (ADaM) is another system that deals with disturbed data mining. It is a data mining toolkit designed for use with scientific and image data. It includes pattern recognition, image processing, optimization, and association rule mining capabilities. It is used to apply data mining technologies to remotely-sensed and other scientific datasets associated with Earth science domains. It is a collection of tools for data mining and image processing.

In different domains, a number of research projects are being carried out with a view of discovering knowledge and rules from large sets of geographically distributed heterogeneous data.

DataMiningGrid Consortium is developing tools and services for deploying data mining applications on the grid. CoreGRID is a grid project funded by the European Commission. It aims at strengthening and advancing scientific and technological excellence in the area of Grid and Peer-to-Peer technologies.

NextGrid is another grid project funded by the European Commission which aims to develop a new architecture for Next Generation Grids which will enable their widespread use business and industry.

Today there exist a number of advanced data mining systems. But the major back draw of most of the system is they are very much domain dependent and complex, and also most of them haven't realized the benefit the P2P technology can bring to this process of knowledge discovery. There is a need to have a system that is generic in form and very simple in implementation. The system we are implement was built with this point in view. The system is generic in nature to a great extent and is also simple to implement.

Although there are a number of systems, most of them are very complex and domain specific. There is a need to have a system that is generic in form and very simple in implementation. The system we are implement was built with this point in view. The system is generic in nature to a great extent and is also simple to implement.

3. A Framework of Distributed Data Mining

For a job of classification, we first need to obtain a classifier from the training data and then use the classifier to classify the unclassified data. Therefore, we first obtain the classifier from the network and then perform the job of classification. It consists of the user node (node which initiates the process), broker node (node which has the computational algorithms and various grid status stored), data nodes (nodes which have the data stored) and the computation nodes (where the classification algorithms are generated). Figure 1 gives the system architecture.

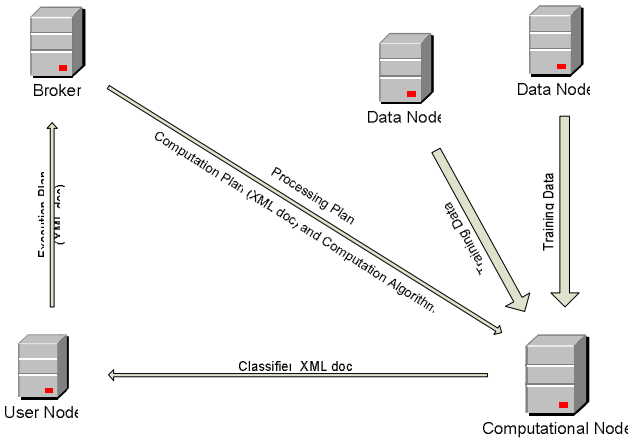

Figure 1. System architecture

3.1. System Requirements

The basic task of the system is to do the job of classification at any node on the grid.

Any node that needs to classify data should be able to generate a plan i.e. and execution plan for the process of generating a classifier. The plan should be able to invoke the data node to transfer the training data to the computation node. The computation node then should be able to generate the necessary algorithm and send it back to the user node. The user node, based on the classifier, should be able to classify the given data.

To generate an execution plan a node needs different kinds of network information. Information required is:

- No. of nodes available
- Resource information for each node; this includes storage and computational resources
- Training data available on the network.

3.2. Data Sets

Training datasets are required for a data mining job. Data mining is a statistical process which requires a lot of training data to discover the rules and patterns in them.

Therefore, we first obtain the classifier from the network and then perform the job of classification. It consists). Figure 1 gives the system architecture.

3.3. Components

As stated above the system mainly consists of 4 main nodes: User node; Broker node; and the Computation node.

User node is the node where the job of classification is to be done. Here the user initiates the whole process. This node is responsible for the planning of the process.

User node creates the execution plan. To create such plan it requests (sends out broadcast messages) resource information from the broker node. Then based on the analysis of the network information it receives from the broker node it creates the plan and sends it back to the broker node. User node (any node in the network should maintain the list of brokers on the network).

Broker node is the one that receives the execution plan and manages the whole process. Broker node receives the information about all the nodes in the network as broadcast message from individual nodes. It then maintains this information and provides to other nodes on request. It also stores computation algorithms. Each broker node broadcasts its status to the other nodes periodically.

Data node is the one that contains the training data. The data are sent to the Computation node for the processing according to the execution plan. Data may be distributed on a number of nodes. Each data node broadcasts its status to the other nodes periodically.

Computation node does the processing of the data from data node. It is here where most of the processing job is done. All the jobs for feature selection, and classifier generation is carried out here. Computation node generates the output (e.g. classifier) and sends it back to the user node. It is also responsible for combining the results in case of more than one computational node. Each data node broadcasts its status to the other nodes periodically.

3.4. Communication

Different components on the network also need to communicate among themselves to carry out the whole process. We have defined the modes and the form of communication.

All the communication on the network is carried out in XML format. We have used simple socket based connection in our implementation of our system. Final results (e.g. classifiers or rules) are also produced in a XML format.

4. Implementation

4.1. Initial Set-up

Once the network is setup and required system components installed in the respective nodes, the system enters running state.

At this stage all the nodes involved broadcast their related information periodically.

This information includes:

- Storage and computational resources information
- Information about the data sets present on the node

Once these messages are broadcast the broker nodes on the network process them and maintain the information within them.

Also, the broker nodes notify their presence to other nodes on the network by means of message broadcast.

On receival of the broadcast messages from the broker node, each user node maintains a table of broker nodes present on the network.

It should be noted that the only predefined nodes on the network are the broker nodes. Other nodes are defined during the process of the mining process based on the analysis of the resources available.

4.2. Mining Process

Once a job (for e.g. classification) is submitted at a user node, the user node initiates the data mining process. Here we present the steps involved in the process.

Step 1:

User node sends out Network Resource Information Request (NRI) to broker nodes listed in its broker node table.

Step 2:

The broker nodes reply to the NRI Request with network resource information.

Step 3:

Once the user node receives the replies (may be

from more than one broker) it analyses the replies and prepares the Execution plan. The plan describes in detail the steps and the components involved in the process. Figure 2 is a sample Execution plan. It also provides the information regarding the data set (training data) to be used and the database access settings.

Step 4:

The execution plan is processed at the broker node. Broker node then sends a processing plan and a computation plan along with the computation algorithm to the computation node.

Step 5:

On the receival of the processing plan, computation plan and the computation algorithm, computation node sends data request to the data node.

Step 6:

Data nodes, on the receival of the data request, preprocess and send the processed data to the computation nodes. Preprocessing may involve simple data cleansing and key attribute selection.

Step 7:

Once the computation node receives the data, it starts generating requested algorithm (for e.g. classifier).

Step 8:

Finally the classifiers are sent back to the user node for classification. User node classifies the unclassified data.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE ExecutionPlan (View Source for full doctype...)>
- <ExecutionPlan>
- <Data_Task epid="EP1">
- <DataAccess>
<dbName>trainingdata</dbName>
<dbTable>dm</dbTable>
<dbUser>prakash</dbUser>
<dbPassword>iselab</dbPassword>
</DataAccess>
- <DataTransfer>
- <Source>
<sHost>220.69.188.106</sHost>
<sPort>4567</sPort>
</Source>
- <Destination>
<dHost>202.6.188.70</dHost>
<dPort>3969</dPort>
</Destination>
</DataTransfer>
</Data_Task>
- <Computation_Task cid="CP1">
<Input>dm</Input>
<Output>EP1.xml</Output>
- <Output_Destination>
<oHost>220.69.185.119</oHost>
<oPort>3456</oPort>
</Output_Destination>
</Computation_Task>
</ExecutionPlan>
```

Figure 2. Execution plan

5. Testing

For the purpose of testing the system we have taken the UCTrainData and UCTestData dataset from UC Data Mining Competition 2005[6]. UCTrainData is the training dataset and UCTestData is the test dataset.

The datasets are Time series of records for 2,528 accounts. Here the job is to determine whether a given account is 'good' or 'bad'. For each account, there is a time series of between 1 and 10,000 records. Each record contains 41 pieces of information. The first value is the account id and the second value is the record id. Values 3 through 41 are data (Boolean, real, integer) associated with each record. The training data also has a 42nd value which is the record label.

System testing went well. We were able to implement the idea of distributed data mining with a very simple system with very few system requirement constraints. Figure 3 and Figure 4 are the screen shots of the system.

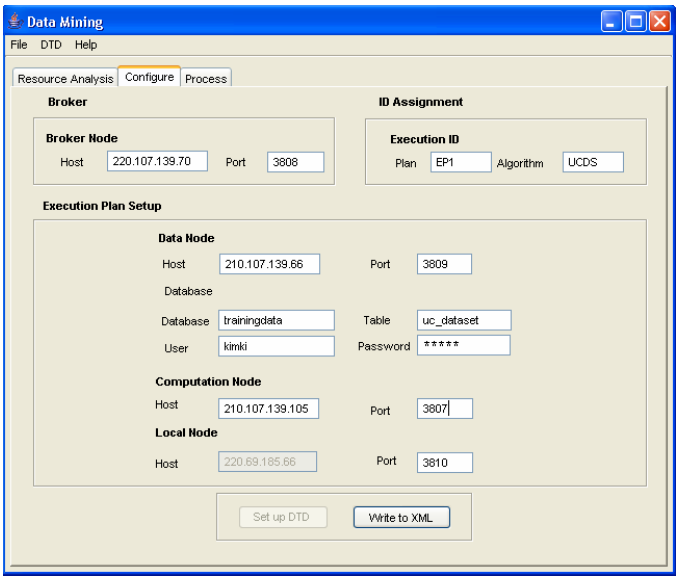

Figure 3. Result of the resource analysis at the broker

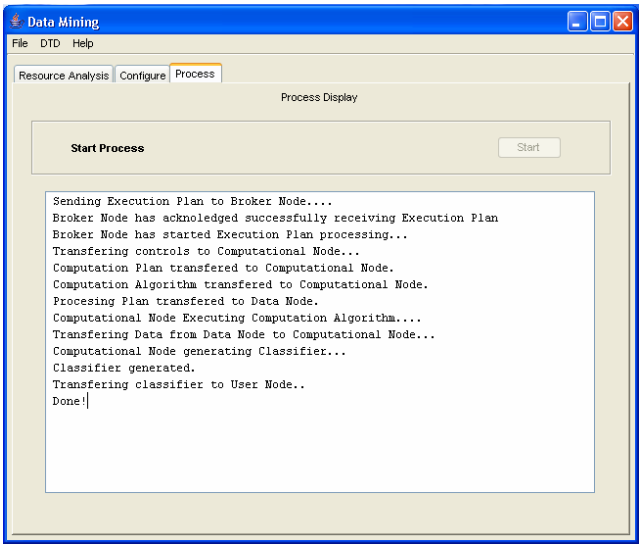

Figure 4. Processing result at the broker

6. Conclusion

In conclusion the system Implementation was successful. We were able to implement the system with much less complexity. The system is very independent with respect to the nature of data mining job the system performs. But there also remain many issues to be solved.

One of the main issues is the issue of maintaining the grid status information in the broker node. Grid status information refers to the state of the resources available on the grid. To maintain such information all related nodes need to broadcast any change in their status to the broker. This may make the grid flooded with node broadcast.

Maintaining single broker on the grid may lead to single point failure.

To generate the classifier, data needs to be transferred to the computation node. Transferring huge amount of data may be time consuming making the whole process inefficient.

6. Acknowledgement

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement).

6. References

[1] M. Cannataro, A. Congiusta, A. Pugliese, D. Talia, P. Trunfio, “Distributed data mining on grids: services, tools, and applications”, IEEE Transactions on Systems, Man and Cybernetics (Part B), Vol. 34, No. 6, pp. 2451-2465, Dec. 2004.

[2] H. Huang, “Enterprise PACS and Image Distribution”, Computational Medical Imaging and Graphics, Vol. 27, No. 2-3, pp. 241-253, 2003.

[3] M. Cannataro, D. Talia, P. Trunfio, “KNOWLEDGE GRID: High Performance Knowledge Discovery Services on the Grid”, Proceedings of the GRID-2001 Conference, LNCS, pp. 38-50, Springer-Verlag, 2001.

[4] D. Skillicorn, “Strategies for Parallel Data Mining”, IEEE Concurrency, Vol. 7, No. 4, pp. 26-35, 1999.

[5] J. Rushing, R. Ramachandran, U. Nair, S. Graves, R. Welch, H. Lin, “ADaM: a data mining toolkit for scientists and engineers”, Computers and Geosciences, Vol. 31, No. 5, pp. 607-618, 2005.

[6] University of California at San Diego, Fair Isaac - University of California Data Mining Competition 2005. [http://mill.ucsd.edu/index.php?page=Main]