ISSN No. 0976-5697

Volume 1, No. 4, Nov-Dec 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Design and Development of Data Mining on Computational Grids in Grid Miner

Dr.P. Sengottuvelan
Assistant Professor
Department of Information Technology
Bannari Amman Institute of Technology
Sathyamangalam-638 401, India
sengottuvelan@rediffmail.com

T. Gopalakrishnan*
Lecturer
Department of Information Technology
Bannari Amman Institute of Technology
Sathyamangalam-638 401, India
mailzone.gopal@gmail.com

Abstract: Data mining algorithms and knowledge discovery processes are both compute and data intensive, therefore the Grid can offer a computing and data management infrastructure for supporting decentralized and parallel data analysis. Distribution of data and computation allows for solving larger problems and execute applications that are distributed in nature. The Grid is a distributed computing infrastructure that enables coordinated resource sharing within dynamic organizations consisting of individuals, institutions, and resources. The Grid extends the distributed and parallel computing paradigms allowing resource negotiation and dynamical allocation, heterogeneity, open protocols and services. Grid environments can be used both for compute intensive tasks and data intensive applications as they offer resources, services, and data access mechanisms. Grid-based data mining uses Grids as decentralized high-performance platforms where to execute data mining tasks and knowledge discovery algorithms and applications. Here we outline the research activities, challenges in the Grid based mining and sketch the promising future directions for developing Grid based distributed data mining. This paper discusses how Grid computing can be used to support distributed data mining.

Keywords: Grid Computing; Distributed Mining; Parallel Computing; Data Analysis; Algorithms.

I. INTRODUCTION

Grid computing represents the natural evolution of distributed computing and parallel processing technologies. The Grid is a distributed computing infrastructure that enables coordinated resource sharing within dynamic organizations consisting of individuals, institutions, and resources [1]. The main aim of grid computing is to give organizations and application developers the ability to create distributed computing environments that can utilize computing resources on demand. Grid computing can leverage the computing power of a large numbers of server computers, desktop PCs, clusters and other kind of hardware. Therefore, it can help increase efficiencies and reduce the cost of computing networks by decreasing data processing time and optimizing resources and distributing workloads, thereby allowing users to achieve much faster results on large operations and at lower costs.

Data mining algorithms and knowledge discovery processes are both compute and data intensive [3]; therefore the Grid offers a computing and data management infrastructure for supporting decentralized and parallel data analysis. The opportunity of utilizing grid based data mining systems, algorithms and applications is interesting to users wanting to analyze data distributed across geographically dispersed heterogeneous hosts. Grid-based data mining would allow corporate companies to distribute compute-intensive data analysis among a large number of remote resources.

A few research framework currently exists for deploying [2] distributed data mining applications in grids. Some of them are general environments supporting execution of data mining tasks on machines that belong to a Grid, others are single mining tasks for specific applications that have been "gratified", and some others are implementations of single data mining algorithms. This paper discusses some approaches for exploiting Grid computing to support

Distributed data mining for water board data by using Grids as decentralized high-performance platforms where to execute data mining tasks and knowledge discovery algorithms and applications.[18].

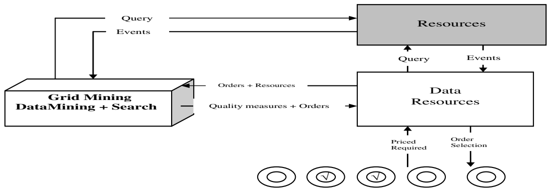

Figure 1: Knowledge grid Framework.

II. DISTRIBUTED DATA MINING AND GRIDS

Today many organizations, companies, and scientific centers produce and manage large amounts of complex data and information [8]. Climate data, astronomic data, water data and company transaction data are just some examples of massive amounts of digital data repositories that today must be stored and analyzed to find useful knowledge in them. This data and information patrimony can be effectively exploited if it is used as a source to produce knowledge necessary to support decision making [5][13]. This process is both computationally intensive and collaborative and distributed in nature. Unfortunately, high-level products to support the knowledge discovery and management in distributed environments are lacking [14].

This is particularly true in Grid-based knowledge discovery [4], although some research and development projects and activities in this area are going to be activated mainly in Europe and USA, such as the Knowledge Grid, the Discovery Net, and the ADAM project. In particular, the Knowledge Grid that provides a middleware for knowledge discovery services for a wide range of high performance distributed applications [6]. Examples of large and distributed data sets available today include gene and protein databases, network access and intrusion data, drug features and effects data repositories, astronomy data files, and data about web usage, content, and structure. Knowledge discovery procedures in all these application areas typically require the creation and management of complex, dynamic, multi-step workflows [7][17]. At each step, data from various sources can be moved, filtered, and integrated and fed into a data mining tool. Based on the output results, the analyst chooses which other data sets and mining components can be integrated in the workflow or how to iterate the process to get a knowledge model. Workflows are mapped on a Grid assigning its nodes to the Grid hosts and using interconnects ions for communication among the workflow components (nodes).

III. OUR APPROACH FOR WATER BOARD

This paper will combine the data mining and grid computing that provides services to the users. Database of water information which is collected from water board of tamilnadu government is used for our implementation .This system is by developing a service oriented grid based frame work that will define, coordinate and manage access to distributed unstructured data[9]. This grid based framework will aid the integrate with core backend data mining services, manage relevant security issues in a distributed environment and allow the integration of distributed resources at application side[15]. This paper is to design and implement data mining applications by using the knowledge grid tools star ting from selecting water types from various locations to final decision making suggestions about water types by using a new decision-tree-based classification algorithm, called SPRINT (Scalable Parallel Classifier for Data Mining)[12] that removes all of the memory restrictions, and is fast and scalable. Here implementing grid computing for this data mining will reduce time of process and successfully complete the processes from distributed databases [16].

IV. SPRINT SPECIFICATION

Classification is an important data mining problem. Although classification is a well-studied problem, most of the current classification algorithms require that all or a portion of the entire dataset remain permanently in memory [8]. This limits their suitability for mining over large data bases. We present a new decision-tree-based classification algorithm, called SPRINT that removes all of the memory restrictions, and is fast and scalable. The algorithm has also been designed to be easily parallelized, allowing many processors to works together to build a single consistent model. This parallelization, also presented here, exhibits excellent scalability as well. The combination of these characteristics makes the proposed algorithm an ideal tool for data Mining [10].

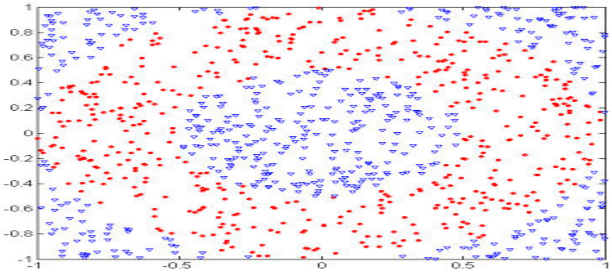

Figure 2: 500 circular and 500 triangular data points for fitting

EvaluateSplits ()

Step 1: for each attribute A do traverse attribute list of A;
Step 2: for each value v in the attribute list do find the corresponding entry in the class list, and hence the corresponding entry class and the leaf node (say l) updates the class histogram in the leaf l;
Step 3: if A is numeric attribute then compute splitting index for test (A ¡= v);
Step 4: if A is a categorical attribute then for each leaf of the tree do find subset of A with best split.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

V. SLIQ ALGORITHM

As many other classic classification algorithms, the SLIQ [11] also can be implemented in two phases: tree building phase and tree pruning phase. Because it is .t for both numerical and categorical attributes, there are a few differences in handling the two kinds of attributes. In the tree building phase, it uses a pre-sorting technique in the tree-growth phase for numerical attributes for evaluating splits while it uses a fast sub setting algorithm for categorical attributes for determining splits. This sorting procedure is integrated with a breadth-first tree growing strategy to enable classification of disk-resident datasets. In the pruning phase, it uses a new algorithm that based on the MDL (Minimum Description Length) principle and gets the results in compact and accurate trees.

A. *Details of the algorithm*

SLIQ algorithm is .t for both numerical and categorical attributes. In this system, the attribute history we will consider are numerical and the others are categorical.

Phase of building tree: In this phase, there are two operations happen. First operation is to evaluate of splits for each attribute and to select the best split; Second operation is partition the training dataset using the best split.

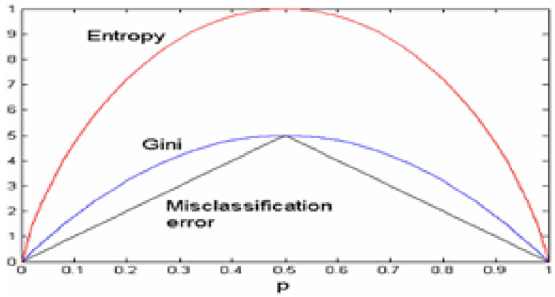

Figure:3 Comparison among Grid Splitting Criteria in Class problem.

The algorithm is described as following:

```
MakeTree(Training Data T)
Partition (T);
Partition (Data S)
If (all records S are in the same class) then return;
Evaluate splits for each attribute A
Use the best split to partition S into S1 and S2;
Partition (S1);
Partition (S2);
```

Before we analysis the numerical attributes, we partition the dataset by attributes -favor and job. Both of them are categorical. Let S (A) is the set of possible values of the attribute A, the split for A is of the form A S', where S' is subset of S. The number of possible subset for an attribute with n possible value is 2 ** n. If the cardinality of S is large, the evaluation will be expensive. Usually, if the cardinality of the S is less than a threshold, MAXSETSIZE (the default value is 10), all of the subsets of S are evaluated. Otherwise, we use the greedy algorithm to get the subset. The algorithm starts with an empty S' and adds one element of S to S' that be the best split, these process will be repeated until there is no improvement in the splits.

B. Design of SPRINT Algorithm

The technique of creating separate attribute lists from the original data was first proposed by the SLIQ algorithm. In SLIQ, an entry in an attribute list consists only of an attribute value and a rid, the class labels are kept in a separate data-structure called a class list which is indexed by rid. In addition to the class label, an entry in the class list also contains a pointer to a node of the classification tree which indicates to which node the corresponding data record currently belongs. Finally, there is only one list for each attribute. The advantage of not having separate sets of attribute lists for each node is that SLIQ does not have to rewrite these lists during a split. Reassignment of records to new nodes is done. simply by changing the tree-pointer field of the corresponding class-list entry. Since the class list is randomly accessed and frequently updated, it must stay in memory all the time or suffer severe performance degradations. The size of this list also grows in direct proportion to the training-set size. This ultimately limits the size.Our goal in designing SPRINT was not to outer form SLIQ on datasets where a class list can fit in memory. Rather, the purpose of our algorithm is to develop an accurate classifier for datasets that are simply too large for any other algorithm, and to be able to develop such a classifier efficiently. Also, SPRINT is designed to be easily parallelizable.

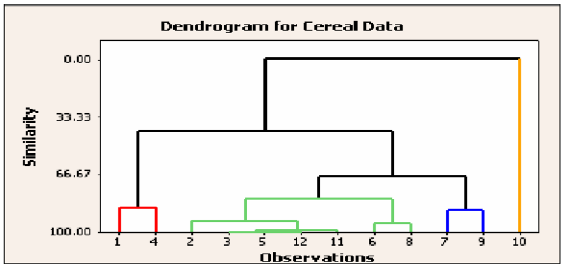

Figure 4:Sample Grid Framework Ouput Model.

C. Parallelizing Classification

We now turn to the problem of building classification trees in parallel. We again focus only on the growth phase due to its data intensive nature. The pruning phase can easily be done off-line on a serial processor as it is computationally inexpensive, and requires access to only the decision-tree grown in the training phase In parallel tree-growth, the primary problems remain finding good split-points and partitioning the data using the discovered split points. As in any parallel algorithm, there are also issues of data placement and workload balancing that must be considered. Fortunately, these issues are easily resolved in the SPRINT algorithm. 'SPRINT was specifically designed to remove any dependence on data structures that are either centralized or memory-resident; because of these design goals, SPRINT parallelizes quite naturally and efficiently.

D. The Knowledge Grid framework

Knowledge Grid framework is a system implemented to support the development of distributed KDD processes in a Grid. Figure 4. Shows the knowledge grid model. It uses basic Grid mechanisms to build specific knowledge disco very services. These services can be developed in different ways using the available Grid environments. This approach benefits from "standard" Grid services that are more and more utilized and offers an open distributed knowledge discovery architecture that can be configured on top of Grid middleware in a simple way.

According to the data sets, 5 levels of scale (10, 20, 30, 40 and 50) were chosen during each DT and knowledge classification. Results (Figure6) showed that the accuracy got the highest value of 80.1% when the data scale is 20m. While the scale increased or decreased, both of the classification accuracy decreased and it would decrease faster as the scale became smaller. However, the number of rules taken by the DT learning kept decreasing as the scale became smaller. To complete the process, a subset of simplified rules is selected for each class in turn. The framework supports distributed data mining on the Grid by providing mechanisms and higher level services for searching resources, representing, creating, and managing knowledge discovery processes, and for composing existing data services and data mining services as structured, compound services, so as to allow users to plan, store, document, verify, share and (re-)execute their applications, as well as manage their output results. Complex decision tree scan be difficult to understand, for instance because information about one class is usually distributed through out the tree.

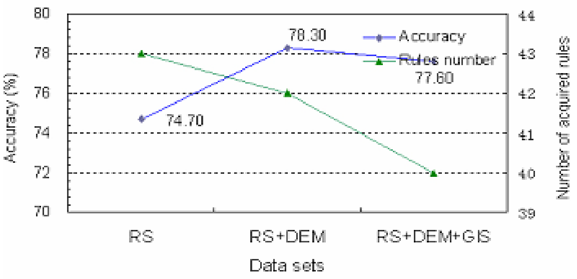

Figure 5: The changing accuracies for the (Decision Tree) DT learning process and knowledge classification using different data sets.

VI. CONCLUSION

Grid can offer an effective infrastructure for deploying data mining and knowledge discovery applications. In the next years the Grid will be used as a platform for implementing and deploying geographically distributed knowledge discovery and knowledge management services and applications. The future use of the Grid is mainly related to its ability embody many of those properties and to manage world-wide complex distributed applications. Among those, knowledge-based applications are a major goal. To reach this goal, the Grid needs to evolve towards an open decentralized infrastructure based on interoperable high-level services that make use of knowledge both in providing resources and in giving results to end users.

VII. REFERENCES

[1] F. Berman. "From TeraGrid to Knowledge Grid", Communications of the ACM, 44(11), pp. 27–28, 2001.

[2] M. Cannataro, D. Talia, "The Knowledge Grid", Communications of the ACM, 46(1), (2003), pp. 89–93.

[3] M. Cannataro, A. Congiusta, C. Mastroianni, A.Pugliese, D. Talia, P. Trunfio, "Grid-Based Data Mining and Knowledge Discovery", In: Intelligent Technologies for Information Analysis, N. Zhong and J. Liu (eds.), Springer-Verlag, chapt. 2 (2004), pp. 19–45.

[4] M. Cannataro, D. Talia, "Semantics and Knowledge Grids: Building the Next-Generation Grid", IEEE Intelligent Systems, 19(1), (2004), pp. 56–63.

[5] K. Czajkowski et al., "The WS-Resource Framework Version1.0."http://www-106.ibm.com/ developerworks/ library/wsresource/wswsrf.pdf.

[6] Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid", In: F. Berman, G. Fox, and A. Hey (eds.), Grid Computing: Making the Global Infrastructure a Reality, Wiley, pp. 217–249, (2003).

[7] H. Kargupta and C. Kamath and P. Chan, "Distributed and Parallel Data Mining: Emergence, Growth, and Future Directions", In: Advances in Distributed and Parallel Knowledge Discovery, AAAI/MIT Press, pp. 409–416, (2000).

[8] D. Talia, P. Trunfio, O. Verta. "Weka4WS: a WSRFenabled Weka Toolkit for Distributed Data Mining on Grids". Proc. PKDD 2005), Porto, Portugal, October 2005, LNAI vol. 3721, pp. 309–320, Springer-Verlag, 2005..

[9] H. Witten and E. Frank. "Data Mining: Practical machine learning tools with Java implementations", Morgan Kaufmann, 2000. vol. 33, no. 2, pp. 293-304.

[10] Rakesh Agrawal, Sakti Ghosh, Tomasz Imielinski, Bala Iyer, and Arun Swami. "An interval classifier for database mining applications", In PTOC. Of the VLDB Conference, pages 560-573, Vancouver, British Columbia, Canada, August 1992.

[11] Rakesh Agrawal, Tomasz Imielinski, and ArunSwami. Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering, 5(6):914-925, December 1993.

[12] Manish Mehta, Rakesh Agrawal, and Jorma Rissanen. "SLIQ: A fast scalable classifier for data mining" In PTOC. of the Fifth Int'l Conference on Extending Database Technology (EDBT), Avignon, France, vol. 14, no. 6, March 1996, pp.735-755.

[13] H. Zhuge, "The Knowledge Grid", World Scientific Publishing Co., Singapore, 2004.

[14] Dietmar W. Erwin and David F. Snelling. UNICORE: A Grid Computing Environment. In Int. Conf. on Parallel and Distributed Computing (Euro-Par'01), volume 2150 of LNCS, pages 825–834, Manchester, UK, 2001, Springer.

[15] Andreas Prodromidis and Philip Chan. "Meta-learning in Distributed Data Mining Systems: Issues and Approaches" In Hillol Kargupta and Philip Chan, editors, Advances in Distributed and Parallel Knowledge Discovery, AAAI press, 2000.

[16] Andrea Pugliese and Domenico Talia. Application-Oriented Scheduling in the Knowledge Grid: A Model and Architecture. In Int. Conf. on Computational Science and its Applications (ICCSA'04), volume 3044 of LNCS, pages 55–65, Assisi, Italy, 2004. Springer.

[17] Giuseppe Bueti, Antonio Congiusta, and Domenico Talia. Developing Distributed Data Mining Applications in the Knowledge Grid Framework. In Int. Conf. on High Performance Computing for Computational Science (VECPAR'04), volume 3402 of LNCS, pages 156–169, Valencia, Spain, 2004. Springer.

[18] Mario Cannataro, Antonio Congiusta, Andrea Pugliese, Domenico Talia, and Paolo Trunfio. Distributed Data Mining on Grids: Services, Tools, and Applications. IEEE Transactions on Systems, Man, and Cybernetics: Part B, 34(6):2451–2465, December 2004

AUTHORS BIOGRAPHY

Dr.P.Sengottuvelan received M.Sc., Degree in Computer Technology from Periyar University, Salem in 2001 and Master of Philosophy in Computer Science from Bharathiar University, Coimbatore in 2003 and M.E. degree in Computer Science & Engineering from Anna University, Chennai in 2004. He also received his Ph.D in degree in Computer Science & Engineering Vinayaka Missions University, Salem in 2010. Since 2004, he has been the Faculty in the Department of IT, BIT, Sathyamanagalam. His current research focuses on Concurrent Engineering, Multi Agent System networks, Constraint Management Agents He is member of IACSIT, ACEEE, IAENG and Life Member of FUWA and ISTE.

T.Gopalakrishnan received B.Tech Degree in Information Technology from Anna University, Chennai in 2005 and M.E. Degree in Computer and Communication from Anna University, Chennai in 2008. Currently he is working as Lecturer in the Department of IT, Bannari Amman Institute of Technology, and Sathyamangalam. He will do part time research in Data Mining at Anna University, Coimbatore. His current research focuses on Software Mining, Business Intelligence Data Mining, and Grid Computing.