Performance Improvement of Association Rule Mining Algorithms through Load Balancing in Distributed Computing Platform

Vidushi Singh[1] and Anil Rajput[2]

[1] Department of IT, Institute of Technology and Science,
Ghaziabad, UP, India.
vidushisingh@its.edu.in

[2] Department of Computer Science and Maths, Govt PG College,
Sehore, MP, India.
dranilrajput@hotmail.com

Abstract

Decrease in hardware costs and increase in computer networking technologies have led to the exponential growth in the use of large-scale parallel and distributed computing systems. One of the biggest issues in such systems is the development of effective techniques/algorithms for the distribution of the processes/load of a parallel program on multiple hosts to achieve goal(s) such as minimizing execution time, minimizing communication delays, maximizing resource utilization and maximizing throughput. The algorithms known as load balancing algorithms, helps to achieve the above said goal(s). The objective of this paper is to identify the challenges of the dynamic load balancing of association rule mining algorithm in the distributed computing environment. In future this work can be extended to analyze the efficiency of existing algorithm and if the existing are not, develop an algorithm for dynamic load balancing of association rule mining (ARM) in distributed computing environment .

Keywords: *Parallel Computing, Distributed Computing, Load Balancing, Association Rule Mining.*

1. Introduction

With the advancement in the technologies including Internet, Ecommerce, artificial intelligence, high performance computing along with sharing of data over geographically distributed sites & various types of applications, every scientific and commercial domain like NASA Earth Observing System (EOS), Wal-Mart, social networking sites (ORKUT FACEBOOK) etc. [1][2][3][4][5] including medium or big sized companies are yielding a huge amount of both structured and unstructured data on the daily basis.

However, in the global competitive environment, keeping in mind the fact that the size of the data almost doubles every 18 months, it is really a big challenge for these companies to organize and further analyze those volumes of data efficiently with flexibility and in a responsiveness manner.

Data warehousing, a structured repository of historical data derived from various data sources meets the informational requirements in the process of knowledge discovery and making quick & correct decisions with continuously increasing data size & dimensions (from gigabytes to terabytes or even larger[16].

Data mining tools, techniques & algorithms efficiently make discoveries of hidden & unpredictable pattern from this pool of data but at the same time the tremendous increase in data size & dimensions also increases the complexity of the data mining algorithms and makes data mining tasks too big & slow to be run on a single processor machine [6]. Might be a single processor not having enough main memory to hold all data. A lot of sequential algorithms definitely fails to meet the scalability requirements in terms of execution times, operating system, I/O and synchronization overheads, therefore enforces the one to use parallel and distributed systems [7]. *Scalability* in general means the ability to obtain the same performance on the same request as the database size increases and this can be achieved by parallel OS, parallel architectures(parallel disks, shared memory, parallel processors etc) as well as parallel DBMS [8].

2. Association Rule Mining

Association rule mining (ARM) is an important research area in the field of data mining. ARM is also an important core data mining technique to discover patterns/rules among items in a large database of variable-length transactions. Efficiency of association rules mining (ARM) has been concerned for several years due to the rapid growth in the worldwide information. The goal of ARM is to identify groups of items that most often occur together. ARM also discover associations between attribute values. Given two distinct sets of attribute values, *X* and *Y*, we say *Y* is associated with *X* if the appearance of *X* implies the appearance of *Y* in the same context ARM outputs a list of association rules of the format *X->Y*, where *X->Y* has a predetermined support and confidence. The prototypical application of ARM is market-basket analysis in which items that are frequently purchased together are identified in order to aid in the layout of items in the store. Apart from the market basket it is also used in graph mining applications like substructure discovery in chemical compounds, Well-known sequential algorithms include Apriori, Eclat, Fp-growth and D-CLUB.

3. Distributed ARM

Distributed ARM(DARM) discovers rules from various geographically distributed data sets. However, the network connection between those data sets isn't as fast as in a parallel environment, so distributed mining usually aims to minimize communication costs. So the goal becomes communication optimization. Data decomposition is very important for distributed memory. Researchers proposed the Fast Distributed Mining algorithm to mine rules from distributed data sets partitioned among different sites.

Therefore, the main challenge for obtaining good performance on distributed mining is to find a good data decomposition among the nodes for good load balancing, and to minimize communication. Distributed ARM algorithms aim to generate rules from different data sets spread over various geographical site hence, they require external communications throughout the entire process [11]. They must reduce communication costs. The main challenges include work-load balancing, synchronization communication minimization, finding good data layout, data decomposition, and disk I/O minimization, which is especially important for DARM[16].

4. Load Balancing of ARM Algorithms in Distributed Computing Environment

On multi computers environment load balancing is a major challenge due to the autonomy of the processors and the interprocess communication overhead incurred in the collection of state information, communication delays, redistribution of load etc. To solve or run distributed or parallel applications, Parallel and distributed computing environment is inherently best choice. In such type of applications, a large process/task is divided and then distributed among multiple hosts for parallel computation.[12] Has pointed out that in a system of multiple hosts the probability of one of the hosts being idle while other host has multiple jobs queued up can be very high. Here load balancing is likely to improve performance Such imbalances in system load suggest that performance can be improved by either transferring jobs from the currently heavily loaded hosts to the lightly loaded ones or distributing load evenly/fairly among the hosts .The algorithms known as load balancing algorithms, helps to achieve the above said goal(s).

In distributed systems load balancing represents allocation or reallocation of task to different processors with the intent of assigning each processor an equal amount of work. The heaviest use of load balancing techniques is found in the domain of distributed systems. However, most of the work is done on computational tasks and not in the storage system area[13].Load balancing techniques are generally widely employed in the domain of distributed system. Most of their application work on reassigning the task between the multiple processor increases computational speed but not the storage area[16].

Work load balancing can be achieved by two ways in distributed ARM algorithms: Static Load Balancing Algorithm and Dynamic Load Balancing Algorithm.

4.1 Static Load Balancing (SLB)

In static load balancing work is initially partitioned among the processors using some heuristic cost function, and there is no subsequent data or computation movement to correct load imbalances which result from the dynamic nature of ARM algorithms.

4.2 Dynamic Load Balancing (DLB)

Dynamic load balancing seeks to address this by stealing work from heavily loaded processors and re-assigning it to lightly loaded ones. Computation movement also entails data movement, since the processor responsible for a computational task needs the data associated with that task as well. Dynamic load balancing thus incurs additional costs for work/data movement, and also for the mechanism used to detect whether there is an imbalance, but it is beneficial if the load imbalance is large and if load changes with time. Dynamic load balancing is especially important in multi-user environments with transient loads and in heterogeneous platforms, which have different processor and network speeds. These kinds of environments include parallel servers, and heterogeneous, meta-, and super-clusters, i.e., the so-called “grid” platforms becoming common today. All extant ARM algorithms use only a static load balancing approach that is inherent in the initial partitioning of the database among available nodes. This is because they assume a zdedicated, homogeneous environment. But in my research work I will work on dynamic load balancing because it is more efficient than static load balancing.

4.3 Hybrid Load Balancing

The third one is **hybrid load balancing** condition when dynamic and static are merge together and perform to take the advantages of both conditions[15].

5. Performance Measurement of Load Balancing Algorithms

On multi computers environment load balancing is a major challenge due to the autonomy of the processors and the inter process communication overhead incurred in the collection of state information, communication delays, redistribution of load etc. To solve or run distributed or parallel applications, Parallel and distributed computing environment is inherently best choice. In such type of applications, a large process/task is divided and then distributed among multiple hosts for parallel computation.[12] Has pointed out that in a system of multiple hosts the probability of one of the hosts being idle while other host has multiple jobs queued up can be very high. Here load balancing is likely to improve performance. Such imbalances in system load suggest that performance can be improved by either transferring jobs from the currently heavily loaded hosts to the lightly loaded ones or distributing load evenly/fairly among the hosts .Load balancing algorithms, helps to achieve the above said goal(s).

High performance data mining refers to parallel data mining techniques and the development of data mining algorithms that can run over a distributed system on parallel computers and can minimize the execution time, I/O overheads etc and can maximize the resource utilization and throughput [9]. In parallel computing, it is very essential to balance the workload equally among all processes with least data dependence across them. Also minimize synchronization, communication overheads and reduce disk I/O cost for the parallel algorithms. Chhabra A. et.al identifies the Processor Thrashing, Preemptiveness, Predictability, Adaptability, Reliability ,stability etc as qualitative parameters for static and dynamic load-balancing algorithms along with the amount of overhead associated due to reallocation of task and inter/intra process communication [10].

The performance of static and dynamic load balancing algorithms can be measured by certain parameters[10][14]:

5.1 Nature

This factor is identifies the behavior of load balancing algorithms, that is whether the algorithm is static or dynamic in naturer no planning. SLB algorithms are planned in nature as tasks are assigned at compile time in a planned manner. In other words we can say that tasks are assigned at compile time to processors and there will be no redistribution of tasks takes place afterwards and outcome of the algorithm is deterministic as much of the job information is known apriori. DLB algorithms are no-planning in nature as tasks are assigned at run-time to processors and tasks redistribution can take place if task assignment that was earlier done is not giving good performance. So their behavior is totally nondeterministic and no initial planning is done for assigning load to hosts as this work is done at run-time [10].

5.2 Overhead Associated

This factor is associated with identification of the amount of overhead incurred while implementing a load-balancing algorithm. It is combination of overhead incurred due to movement of tasks, inter-processor communication, and inter-process communication. SLB algorithms incurs lesser overhead as compare to the DLB because once tasks are assigned to processors in SLB, no redistribution of tasks takes place, so no relocation overhead.

In DLB algorithms relocation of tasks takes place, so it incurs more overhead relatively.

5.3 Resource Utilization

This factor is used to check the resource utilization. SLB algorithms have lesser resource utilization as static load balancing methods just tries to assign tasks to processors in order to achieve minimize response time ignoring the fact that may be using this task assignment can result into a situation in which some processors finish their work early and sit idle due to lack of work. DLB algorithms have relatively better resource utilization as dynamic load balancing take care of the fact that load should be equally distributed to processors so that no processors should sit idle[10].

5.4 Processor Thrashing

Processor thrashing occurs when most of the processors of the system are spending most of their time migrating processes without accomplishing any useful work in an attempt to properly schedule the processes for better performance. SLB algorithms are free from Processor thrashing as no relocation of tasks place. DLB algorithms incurs substantial processor thrashing[10].

5.5 Preemptiveness

This factor is related with checking the fact that whether tasks in execution can be transferred to other nodes (processors) or not. SLB algorithms are inherently non-preemptive as no tasks are relocated. DLB algorithms are both preemptive and non preemptive[10].Mostly DLB algorithms are preemptive in nature.

5.6 Predictability

This factor is related with the deterministic or nondeterministic factor that is to predict the outcome of the algorithm. SLB algorithm's behavior is predictable as most of the things like average execution time of processes and workload assignment to processors are fixed at compile-time. DLB algorithm's behavior is unpredictable, as everything has been done at run time[10].

5.7 Adaptability

This factor is used to check whether the algorithm is adaptive to varying or changing situations i.e. situations which are of dynamic nature. SLB algorithms are not adaptive towards all circumstances as this method fails in dynamic or varying nature problems i.e. situation in which number of processes are not fixed, also in situations which may require indeterminate steps towards solution. DLB algorithms are adaptive towards every situation whether numbers of processes are fixed or varying one[10].

5.8 Reliability

Which algorithm is more reliable in case of some host failure occurs? SLB algorithms are less reliable because no task/process will be relocated / transferred to another host in case a node fails at run-time. DLB algorithms are relatively more reliable as here processes can be transferred to other nodes in case of failure of node occurs[10].

5.9 Response Time

How much time a distributed system using a particular load balancing algorithm is taking to respond? SLB algorithms have shorter response time as one should not forget that in SLB there is lesser overhead as discussed earlier so emphasis is totally on executing jobs in shorter time rather than optimally utilizing the available resources. DLB algorithms may have relatively higher response time as sometimes redistribution of processes takes place. Some time is being consumed during task migration

5.10 Stability

Stability can be related to the exchange of present workload state information among processors. SLB algorithms in this context can be considered as stable as no information regarding present workload state is passed among processors. However in case of DLB such kind of information is exchanged among processors and if this information is out of date i.e. information which is not updated regularly of periodically among processors then it can lead the whole system to an unstable state.

5.11 Issues

Considering other major issues related with load balancing algorithms. SLB algorithms-The major issue in static load balancing is to accurately determining the process execution times, communication delays and other resource needs of a processor a priori. A prior accurate estimation is not possible in reality, so emphasis can be done on to estimation of such quantities close to accurate value. DLB algorithms- The major issue concerning DLB algorithms is to develop fast methods for distributed termination detection and to develop techniques of reducing overhead which includes inter processor communication overhead and task migration overhead, which is main problem in dynamic load balancing.

Table 1: COMPARISON OF ALGORITHMS ON DIFFERENT PARAMETERS

Load Balancing Parameters	**SLB Algorithms**	**DLB Algorithms**
1. Nature	Static i.e. workload is assigned at compile time	Dynamic i.e. workload is assigned at run time
2.Associated overhead	Lesser overhead	More overhead
3.Resource Utilization	Lesser Utilization	More Utilization
4.Processor Thrashing	No Thrashing	Substantial Thrashing
5.Preemptiveness	Non-preemptive	Preemptive and Nonpreemptive
6.Predictability	More Predictable	Lesser predictable
7.Adaptability	Less adaptive	More Adaptive
8.Reliability	Less	More
9.Response Time	Less	More
10.Stability	More	Less
11.Other Issues	Determining process execution time at run time	Developing techniques to reduce Communication overhead

6. *Conclusion*

In order to ensure good overall performance, Load balancing algorithm tries to balance the total system load by transparently transferring the workload from heavily loaded nodes to lightly loaded nodes.
In SLB lesser over head is required but overhead increases in DLB. That overhead will be compensating in resource utilization. DLB utilizes resources more as compare to SLB. Thrashing degrades the performance. There is no thrashing in SLB but substantial thrashing is there in DLB.DLB algorithms are more reliable and adaptable as compare to SLB. Apart from all these parameters SLB algorithms are more reliable works in less response time .

The above comparison shows that static load balancing algorithms are more stable in compare to dynamic and it is also ease to predict the behavior of static, but at a same time dynamic distributed algorithms are always considered better than static algorithms

7. *Future Work*

Future work will be based on comparative study of dynamic load balancing of association rule mining algorithms in distributed environment and their efficiency will be checked. If we find any algorithm efficient then that will be used. But due to any circumstances if any algorithm doesn't find efficient then new efficient algorithm will be developed for dynamic load balancing of ARM algorithm .

REFERENCES

[1] http://esdis.eosdis.nasa.gov/eosdis/overview.html
[2] http://www.internetnews.com/dev-news/article.php/3421501/Wal-Mart-Expands-Data-Warehouse.htm
[3] http://www.encyclopedia.com/doc/1G2-3401200510.html
[4] https://www.blackhat.com/presentations/bh-usa-07/Patton/Whitepaper/bh-usa-07-patton-WP.pdf
[5] http://kdl.cs.umass.edu/papers/jensen-neville-nas2002.pdf
[6] Alexis Berson, Stephen J. Smith , Data Warehousing, Data mining & OLAP, Tata McGraw hill edition
[7] Mohr B., Introduction to Parallel Computing, Computational Nanoscience: Do It Yourself!, NIC Series, Vol. 31, ISBN 3-00-017350-1, pp. 491-505, 2006
[8] Buzbee B.L., The Efficiency of Parallel Processing, Frontiers of Supercomputing, LOS ALAMOS SCIENCE, 1983

[9] Zomaya A.Y, El-Ghazawi T., Frieder O., Parallel and Distributed Computing for Data Mining,IEEE Concurrency, oct-nov, 1999

[10]Amit Chhabra, Gurvinder Singh, Sandeep Singh Waraich, Bhavneet Sidhu, and Gaurav Kumar," Qualitative Parametric Comparison of Load Balancing Algorithms in Parallel and Distributed Computing Environment", World Academy of Science, Engineering and Technology 16 2006

[11] H. Ravi Shankar, M.M. Naidu "An innovative algorithm for mining multilevel association rules", Proceedings of 25th conference on IASTED International Multiconference on Artificial Intelligence and applications pp. 307-310, Innsbruck, Austria, feb 12-14,2007.

[12] Miron Livny, Myron Melman,"Load balancing in homogeneous broadcast distributed systems", Proceedings of the Computer Network Performance Symposium, p.47-55, April 13-14, 1982, College Park, Maryland, United States.

[13] allab Dasgupta, A.K. Majumder, and P. Bhattacharya, "V_THR: An Adaptive Load Balancing Algorithm", Journal of Parallel and Distributed Computing 42, 1997, 101-108.

[14] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma," Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology 38, 2008,269-272.

[15] Zubair Khan, Ravendra Singh, Jahangir Alam and Shailesh Saxena," Classification of Load Balancing Conditions for parallel and distributed systems", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 1, September 2011.

[16] Vidushi Singh, Nancy Sharma and Anil Rajput," Challenges of Dynamic Load Balancing of Association Rule Mining Algorithms in Distributed Computing Platform" International Journal of Data Warehousing and Mining (IIJDWN) Vol(1), issue (1),August 2011.

Biographies

First Author: Vidushi Singh is B.Sc. (1997), M.Sc. (Computer Science) (1999), C-DAC(2001) and pursuing P.hD. in Computer Science from Barkatullah University, Bhopal. She worked in S.G.T.B. Khalsa College Jabalpur (MP) for five and half years. Currently she is working in

Institute of Technology and Science(ITS),Ghaziabad from july 2008.She has published 6 research papers in international journals, international and national conferences. Her current research area is data mining.

Second Author: Dr. Anil Rajput is M. Sc. (Maths), MCA , M. Phil. (CS) and PhD. He worked in Sadhu Vaswani College, Bhopal and also he was Principal in Bhabha Engineering Research Institute, Bhopal. Presently he is Professor in CSA Govt PG College, Sehore. He has published more than 80 papers in international journals, international and national conferences. He has published 4 books.