International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 11 6102 - 6104

Parallel Implementation of Apriori Algorithm on Multicore System for Retail Market

Nitin Maul
Computer Department,
Pimpri Chinchwad College of Engineering, Pune, India
maulnitin@gmail.com

Ganesh Jagtap
Computer Department,
Pimpri Chinchwad College of Engineering, Pune, India
ganeshjagtap967@gmail.com

Pooja Kale
Computer Department,
Pimpri Chinchwad College of Engineering, Pune, India
poojakale115@gmail.com

Priti Waydande
Computer Department,
Pimpri Chinchwad College of Engineering, Pune, India
priti43waydande@gmail.com

Prof. Meghna Lokhande
Computer Department,
Pimpri Chinchwad College of Engineering, Pune, India
meghna.ingole1983@gmail.com

***Abstract*—** Data Mining is a process of examining data and revealing the interesting patterns which are hidden. Association Rule Mining is a key technique of data mining. This technique works on finding intriguing relationships. Association rules are generated using Apriori Algorithm. The set of data includes a number of items which are called transactions. The work of this algorithm is to produce frequent itemsets from the transactional databases based upon the minimum support value. The outcome of an Apriori Algorithm is sets of association rules that provide us the frequency of items that are contained in sets of data which provide us the hidden pattern and general trends. This will be helpful for the retailer be familiar with the market and customer's purchasing behavior. In pursuance of finding more valuable rules, our basic aim is to implement Apriori Algorithm using multithreading approach which can utilization our multicore processing system to improve the performance of an algorithm in a practical and efficient way to unearth more value information for the proper analysis of the business trends.

Keywords- *apriori algorithm; association rules, data mining, parallel apriori algorithm; parallel implementation.*

*****

I. INTRODUCTION

The transactions carried out in the retail sector have the huge amount of data. This data require data mining tool to extract hidden behaviors which may assist an organization to anticipate future trends and behavior of the consumers. Data mining is an enhanced method which refers to the extraction of earlier unidentified and valuable information out of large databases. Association Rule Mining is a vital technique of data mining to discover the unknown pattern which may help to grow the business. This technique emphasis on finding interesting relationships. To understand the customer's purchasing behavior more easily, a technique called Market Basket Analysis can be utilized in Data Mining. Apriori algorithm is used in association rule mining for finding frequent patterns. This paper shows that how the utilization of processing cores improves the efficiency of Apriori algorithm. There will be a comparison against the results of the serially implemented algorithm. The improved algorithm will utilize the multiple cores of the processor for finding the association among the itemsets. This paper shows that how the utilization of processing cores improves the efficiency of Apriori algorithm against the results of the serially implemented algorithm. The improved algorithm will utilize the multiple cores of the processor for finding the association among the itemsets. In the era of a multicore processor we can, increase the performance of an application with off-the-shelf hardware.

II. LITERATURE SURVEY

Today retailers are keen in finding the purchasing habits of the shoppers which in turn will certainly assist them in business associated application such as marketing promotions, stock management and customer relationship management. The traditionally methods consumed lot of time to resolve the problems or decision making for a profitable business. Data mining techniques can scan the databases for discovering disguised patterns that individual may miss. Hence, this paper reviews the various trends of data mining and how effectively can be used for targeting profitable customers in campaigns and utilize the multiple cores of the processor for faster execution.

A. Usage of Apriori Algorithm in retail sector

Jugendra Dongre, S. V. Tokekar, and Gend Lal Prajapati et al. [1], proposed the task of Apriori Algorithm for finding the Association Rules. Association rule mining is interested in finding frequent rules that define relations between unrelated frequent items in databases It includes two major measurements support and confidence values. The frequent itemsets are defined as the itemset that have support value greater than or equal to a minimum threshold support value, and frequent rules as the rules that have the confidence value greater than or equal to minimum threshold confidence value. Typically, association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold. These threshold values are usually

presumed to be readily available for mining recurrent itemsets. Association Rule Mining is to discover all rules whose support and confidence are greater than the threshold, minimum support, and minimum confidence values. Association rule mining basically have two main steps.

i. To find all itemsets with adequate support count.
ii. To generate association rules by combining these frequent or large item-sets.

Associations denoted as A→B. If A is true then B will also true.
Support is the percentage of the population which satisfies the rule or in the other words the support for a rule R is the ratio of the number of occurrence of R, given all occurrences of all rules. The support of an association pattern is the percentage of task- relevant data transactions for which the pattern is true.
Support (A→B) = P (A U B)

$$support(A \rightarrow B) = \frac{number\ of\ tuples\ containing\ both\ A\ and\ B}{total\ number\ of\ tuples}$$

If the percentage of the population in which the antecedent is satisfied is s, then the confidence is that percentage in which the consequent is also satisfied. The confidence of a rule A→B is the ratio of the number of occurrences of B given A, among all other occurrences given A. Confidence decide the measure of truthfulness associated with every uncovered pattern A→B.
Confidence (A→B) = P (B|A) means the probability of B that all know A.

$$confidence(A \rightarrow B) = \frac{number\ of\ tuples\ containing\ both\ A\ and\ B}{number\ of\ tuples\ containing\ \ A}$$

Apriori algorithm was proposed by R. Agrawal and R. Srikant for mining frequent itemsets for discovering Boolean association rules [17]. Apriori makes use of an iterative strategy making the several passes over the database. During first pass database is scanned to count the frequency of individual items and create a list of the items which satisfy the minimum support. This list is called as a set of frequent 1-itemset (L_1). In succeeding itemset generation L_1 is used to generate the set of frequent 2-itemsets L_2, and so on till no new itemsets a generated. Hence, it also known as level-wise search.

B. Multithreaded Java Applications on Manycore Systems

Junjie Qian, Du Li, Witawas Srisa-an et al. [2], Java applications can use multithreading to achieve higher performance by utilizing multiple processing cores. It is important that these applications scale well; that is, the performance of an application should increase as more threads and more processing cores are employed. This performance enhancement continues to occur until the time spent on the sequential part of the program outweighs the enhancement attained through parallelism. Because Java is a managed programming language, the performance of a Java application is determined by two performance factors:

i. The time spent in application execution.
ii. The time spent in execution of runtime systems such as garbage collection (GC time).

Furthermore, these couple of aspects can also impact the scalability of a Java application.
In this paper, they have carried out an investigation to reveal factors that can affect the scalability of Java applications. Their study takes into account together mutator and GC times can lead to the overall scalability of an application. First, they measured application level lock contention with various amounts of threads. The outcomes indicate that for scalable applications, lock contention raises with the number of threads but for improperly scalable applications, lock contention stays basically consistent as the number of threads increases. This implies that performance improvement attained by means of parallelism in scalable applications outweighs the overhead due to higher instances of lock contention. Parallel execution can improve performance by having more threads together performing work. However, such alliance also makes threads contest for resources for example processors and heap space. In scalable applications, the workload can be divided evenly among threads, and therefore, it is advantageous to utilize more threads. However, doing so requires heedful synchronization of shared resources that can result in more lock acquisitions and instances of contention. In addition, it also shows that object lifetime is also affected as the heap usage is an aggregation of objects needed by both active and suspended threads. According to the results reported in this work, to improve the scalability of Java applications by focusing on JVM and OS implementation there are two suggestions.

i. We can bias schedule to minimize lifetime interference that is worker threads are scheduled at the various stages of the execution to cut down contests for heap and locks.
ii. We can make compartmentalized heap to isolate objects from lifetime interference which can most likely enhance throughput performance in large multi-threaded server applications by minimizing memory requirement and shortening garbage collection pause time.

C. Multithreaded Content-Based File Chunking

Youjip Won, Kyeongyeol Lim, and Jaehong Min et al. [3], for partitioning a file into chunks there are two approaches:

i. Fixed size chunking.
ii. Variable size chunking.

Fixed Size Chunking: A file is partitioned into fixed size units, e.g., 16 Kbyte blocks. It is simple, fast, and computationally very cheap.

Variable Size Chunking: In this method partitioning of a file is based on the content of the file, not the offset. Variable size chunking is relatively robust against the insertion/deletion of the file. For variable size chunking, Basic Sliding Window algorithm is widely used.
In this paper, a novel multicore chunking algorithm, MUCH, which parallelizes the variable size chunking. Incorporating this technology advancement, i.e., an increase in the number of CPU cores and the emergence of faster storage devices, we developed a parallel chunking algorithm, which aims at making the variable chunking speed on par with the storage I/O bandwidths. It was found that the legacy variable size chunking algorithm yields a different set of chunks if the parallelism degree changes, a phenomenon referred to as Multithreaded Chunking Anomaly. A multicore chunking algorithm, MUCH, which guarantees Chunking Invariability. The developed performance model is to compute the segment size that maximizes the chunking bandwidth while minimizing the memory requirement. Through extensive physical experiments,

showed that the performance of MUCH scales linearly with the number of cores. In quad-core CPUs, MUCH brings a 400 percent performance increase when the storage device is sufficiently fast. The benefits of MUCH are evident when it chunks large files. MUCH successfully increases the chunking performance with the factor being as high as the number of available CPU cores without any additional hardware assistance.

III. PROPOSED SYSTEM

The multicore processor can make us achieve High-Performance Computing results in Real Time. With the advent of multicore processors and more advanced operating systems that utilize symmetric multiprocessing that allow the software to be load balanced over the multiple cores of a processing unit. In today's world, we can find up to 16 cores in CPU and in near future CPU will have 32 cores. Also, Intel has research program called Intel Tera-Scale which focuses on development in Intel Processors to utilize the multiple cores of the processor to exhibit parallelism. Under this research program, Intel made their first effort at developing a Tera-Scale processor called Teraflops Research Chip (Polaris) which is an 80 core prototype. It gives us the idea that multicore processor has the potential to become a miniature supercomputer.

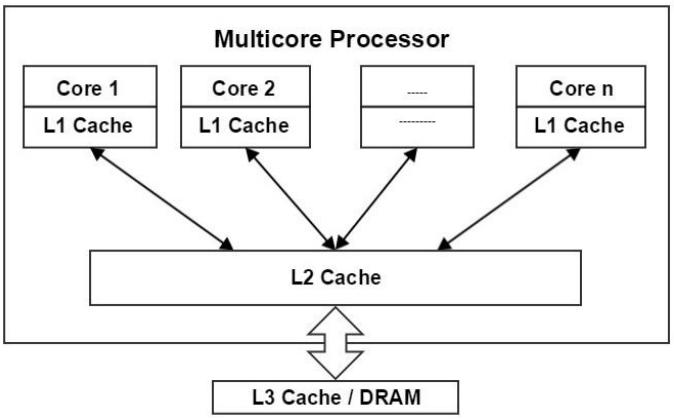

Fig 1. Multicore Architecture.

In this paper, we have proposed the parallel implementation of Apriori Algorithm which can be used by the retailer to extract knowledge for the taking the better business decision.
The transactional database is divided into chunks depending on the cores available on the processor. The chunks are processed by the processor parallel to reduce the execution time. The proposed system is designed to generate result faster with off-the-shelf hardware.

IV. CONCLUSION

In summary, the parallel execution can improve the performance by having more threads. The workload is divided among the multiple cores of the processor thus reducing the execution time. In multicore era, there will be new processing capabilities for mining and discovering association.

V. REFERENCES

[1] Jugendra Dongre, S. V. Tokekar, and Gend Lal Prajapati, "The Role of Apriori Algorithm for Finding the Association Rules in Data Mining" International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT),IEEE Catalogue Number: CFP1463W-DVD ISBN: 978-1-4799-2899-6, 2014.
[2] Junjie Qian, Du Li, Witawas Srisa-an, Hong Jiang and Sharad Seth, "Factors Affecting Scalability of Multithreaded Java Applications on Manycore Systems", 2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS) 2015, pp. 167-168, doi:10.1109/ISPASS.2015.7095800, 2015.
[3] Youjip Won, Kyeongyeol Lim, and Jaehong Min. "MUCH: Multithreaded Content-Based File Chunking", IEEE Transaction on Computers, VOL. 64, NO. 5, ISSN: 0018-9340, May 2015,
[4] N. Baddal, S. Bagga, "Implementation of Apriori Algorithm in MATLAB using Attribute Affinity Matrix" IJARCSSE, Vol. 4, Issue 1, ISSN: 2277 128X, Jan 2013.
[5] Jaishree Singh*, Hari Ram**, Dr. J.S. Sodhi, "Improving the efficiency of Apriori algorithm by transaction reduction", International Journal of Scientific and Research Publications, Volume 3, Issue 1, ISSN 2250-3153, January 2013.
[6] Anshuman Singh Sadh, Nitim Shukla, "Apriori and Ant Colony Optimization of Association Rules", International Journal of Advanced Computer, Volume-3 Number-2 Issue-10, ISSN: 2249-7277, June-2013.
[7] Sheila A. Abaya, "Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation", In: International Journal of Scientific & Engineering Research Volume 3, Issue 7, ISSN 2229-5518, July-2012
[8] Lingjuan Li, Min Zhang, "The Strategy of Mining Association Rule Based on Cloud Computing", 2011 International Conference on Business Computing and Global Informatization IEEE, Print ISBN: 978-1-4577-0788-9.
[9] Mamta Dhanda, "An Approach to Extract Efficient Frequent Patterns from Transactional Database", In: International Journal of Engineering Science and Technology (IJEST), Vol.3 No.7 July 2011, ISSN: 0975-546.
[10] J. Hossen, A. Rahman, K. Samsudin, F. Rokhani, S. Sayeed, and R. Hasan, ― A Novel Modified Adaptive Fuzzy Inference Engine and Its Application to Pattern Classification, World Academy of Science, Engineering and Technology 80, 2011
[11] Goswami D.N., Chaturvedi Anshu, Raghuvanshi C.S., "An Algorithm for Frequent Pattern Mining Based On Apriori", In: Goswami D.N. et. al. / (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 04, 2010, 942-947, ISSN: 0975-3397
[12] Yan Zhang, Jing Chen, "AVI: Based on the vertical and intersection operation of the improved Apriori algorithm", Future Computer and Communication (ICFCC), 2nd International Conference on (Volume: 2), Page(s): V2-718 - V2-721, Print ISBN: 978-1-4244-5821-9, 2010.
[13] Yanxi Liu, "Study on Application of Apriori Algorithm in Data Mining" Computer Modeling and Simulation, 2010. ICCMS '10. Second International Conference on (Volume: 3), Page(s): 111 – 114, Print ISBN: 978-1-4244-5642-0
[14] Changsheng Zhang and Jing Ruan "A Modified Apriori Algorithm with Its Application in Instituting Cross-Selling Strategies of the Retail Industry". 2009, International Conference on Electronic Commerce and Business Intelligence, Print ISBN: 978-0-7695-3661-3
[15] J. Han, M. Kamber, Data mining: Concepts and Techniques, Morgan Kauffman, San Francisco, 2000, ISBN: 1-55860-489-8
[16] Zaki MJ (1999) Parallel and distributed association mining: a survey. Concurrency, IEEE (Volume: 7, Issue: 4), Page(s): 14–25, Special issue on Parallel Mechanisms for Data Mining, ISSN: 1092-3063
[17] Agrawal R, Srikant R - Fast algorithms for mining association rules‖ In: Proceedings of the 1994 international conference on very large data bases (VLDB‟94), 1994 Santiago, Chile, and pp 487–499, ISBN: 1-55860-153-8
[18] From a Few Cores to Many: A Tera-scale Computing Research Review White Paper, Research at Intel, www.intel.com/go/terascale