IOSR Journal of Computer Science (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727
PP 01-04
www.iosrjournals.org

Parallel Implementation of Apriori Algorithm

Nilesh.S.Korde[1], Prof.Shailendra.W.Shende[2]
[1]*(Department of Information Technology, Yeshwantrao Chavan College of Engineering Nagpur,India)*
[2]*(Department of Information Technology, Yeshwantrao Chavan College of Engineering Nagpur, India)*

***ABSTRACT** : Association rule mining concept is used to show relation between items in a set of items. Apriori algorithm for mining frequent itemsets from large amount of database is used. Parallelism is used to reduce time and increase performance, Multi-core processor is used for parallelization. Mining in a Serial manner can consume time and reduce performance for mining. To solve this issue we are proposing a work in which load balancing is done among processors. Our work is based on serial and parallel implementation of Apriori algorithm and comparison of both on the basis of time and varying support-count. For parallelization most widely used parallel programming technique Open Multi-Processing (OpenMP) can be used.*
***Keywords** –Apriori, Data mining, Multi-core processor, OpenMP, Parallel processing*

I. INTRODUCTION

Data mining is a technique for mining hidden knowledge from large number of databases, knowledge which is never seen before [1]. For knowledge discovery data mining is used. Association rule mining (ARM) is one of the functionality of data mining. Association rule mining is used to find relation between two items, which are frequently appearing with each other in a large database. Apriori algorithm is one that is widely used for mining frequent itemsets from huge transactional databases. Apriori algorithm for finding frequent itemsets scans database more than once [1]. Dealing with large volume of data, data mining requires increase in performance of algorithm. Parallelizing Apriori algorithm can improve performance of Apriori algorithm drastically [2]. For parallelizing Apriori algorithm multi-core architecture is required, which is provided by recently launched advanced computers. In advanced computers multi-core processor is available [3]. Multi-core processor shares the load among cores with single thread or multiple threads, which reduce time for processing and increase performance [4].

II. RELATED WORK

Rakesh Agrawal and Ramakrishna srikant presents two new algorithms Apriori and ApriroriTid and combined into an Apriori hybrid algorithm and demonstrate in scale-up-properties [5]. Chao Yang, Tzu Chang and Chih Chang presents the comparison of some tools that are specifically designed to extract the most of data parallelism on multi-core system using OpenMP [3].Ying Liu and Fuxiang Gao presents the cubic convolution interpolation algorithm for image processing and parallelized it by using OpenMP on multi-core processors [4]. Ketan shah and Sunita Mahajan present the performance of parallel Apriori algorithm on heterogeneous nodes with different datasets and n processors on a commodity cluster of machines [6]. Anuradha.T and Satya Prasad.R presents an evaluation of the performance of Apriori on a hyper threaded dual core processor compared to the performance on a non hyper threaded dual core processor using fread() and mmap() functions [7]. Zhang Zheng, Jaiwen Li, Xuhoo Chen, Li Shen, and Zhiying Wang present a performance model for OpenMP parallelized loops to address the critical factors which influences the performance [8].Kyung Min Lee, Tae Houn Song, Seung Hyun Yoon, Key Ho Kwon and Jae Wook Jeon presents a parallel programming model, OpenMP, and parallel programs that can be benchmarked to multi-core processors of embedded boards using OpenMP and Executed parallel programs on a dual-core embedded system, analyzing the performance of sequential programs and parallel programs by SERPOP analysis [9].Anuradha.T, Dr.Satya Prasad.R, Dr.Tirumala Rao.S.N, evaluates the performance of Apriori using Linux mmap() function compared to fread() function in both the serial and parallel environments [10].

III. THEORETICAL BACKGROUND

A. Association RuleMining:

The concept of Association rule mining was originated from the market basket analysis. It considers a transactional database D which consists of the transactional records of the customers {T1, T2… Tn}. Each transaction T consists of the items purchased by the customers in one visit of the super market. The items are the subset of the set of whole items I in the super market we are considering for analysis. We represent I as the set {I1, I2,….,Im}.An itemset consists of some combination of items which occur together or a single item from I. Association rule mining X Y, represents the dependency relationship between two different itemsets X and Y in the database. The relationship is whenever X is occurring in any transaction, there is a probability that Y may also occur in the same transaction. This occurrence is based on two interesting measures [1].

1. Support= percentage of transactions in D that contain XUY $supportX = P\ (X \cup Y)$
2. Confidence=percentage of transactions in D containing X that also contain Y. $confidenceXY = P\ (YX)$

Finding the association rules for any given transactional database consists of two parts:

1. Finding the frequent itemsets— Frequent itemsets are the itemsets which are having a frequency more than a predefined minimum support count (min_sup).
2. Generation of Association rules— Rules generated from the subsets of a frequent itemset [1].

B. Apriori Algorithm:

It is nothing but finding frequent itemsets using candidate generation. It uses Apriori property that all nonempty subsets of a frequent itemset must also be frequent. Two steps used in Apriori algorithm are

Step 1: The Prune Step: The entire database is scanned to find the count of each candidate in Ck. Ck represents candidate k-itemset. The count each itemset in Ck is compared with a predefined minimum support count to find whether that itemset can be placed in frequent k-itemset Lk [1].

Step 2: The join step: Lk is natural joined with itself to get the next candidate k+1- itemset Ck+1.

The major step here is the prune step which requires scanning the entire database for finding the count of each itemset in every candidate k-itemset. So to find all the frequent itemsets in the database, it requires more time if the database size is more [1].

C. OpenMP:

OpenMP (Open Multi-Processing) is an Application programming interface (API) which is used for shared memory parallel programming .This specification provides a model for parallel programming that is portable across shared memory architecture from different vendors [11]. Compilers from numerous vendors support the OpenMP API. There primary components of the API are compiler directives, Runtime library routines and Environment variables [12]. OpenMP supports multi-platform shared memory multiprocessing programming in JAVA, C, C++ and FORTRAN on much architecture, including UNIX and Microsoft Windows platforms [13]. OpenMP provides a task construct useful for the expression of unstructured parallelism and for defining dynamically generated unit of work [13]. The advantage of OpenMP is its comparative simplicity of use, because the detailed working for parallel program is up to the compiler [12].

D. Multi-core processor:

Multi-core processors are those having two or more cores integrated on a single chip[14]. The core can be with one thread or two thread each [14]. For parallelism Multi-core processor plays very important role as the cores share work and helps to perform load balancing. The performance of multi-core processor depends on the software used. In load balancing the work is equally shared among the processors which minimizes the execution time and improve performance [14].

I. PROPOSED WORK

We are proposing aserial and parallel implementation of Apriori algorithm on a quad core prosessor. Our objectives are measuring the serial and parallel performance of Apriori agorithm on aquad core processor with respect to time and comparison between them. For our proposed work we mainly used intel core 2 quad processor, Ubuntu 12.10 Linux operating system and five standard datasets from

Frequent Itemset Mining Implementations Repository.We have used different support counts from 40% to 80% for measuring performance.

We have implemented Apriori algorithm on a quad core processor in a serial manner with five standard datasets using different support counts varing from 40% to 80%. The Real time observations are as follows

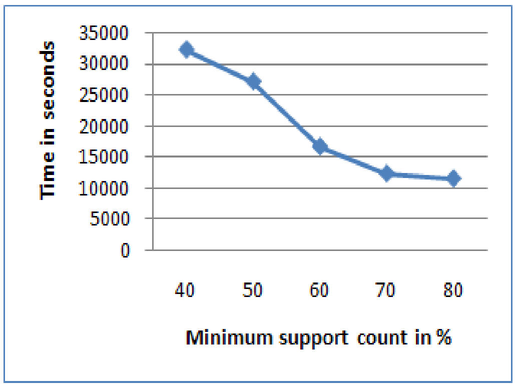

Fig 1: Real time values for ACCIDENTS dataset

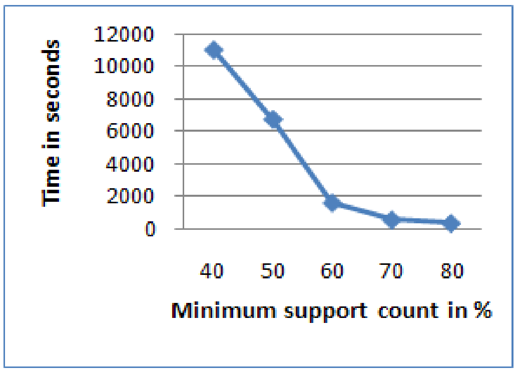

Fig 2: Real time values for RETAIL dataset

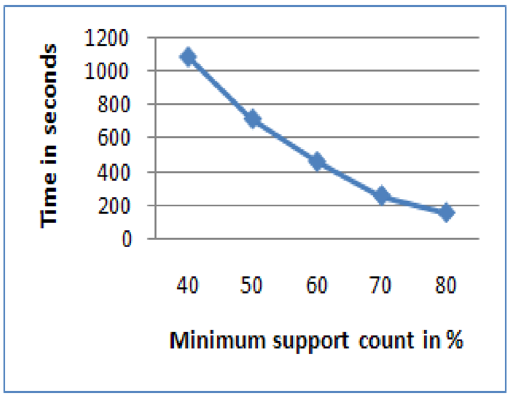

Fig 3: Real time values for MUSHROOM data

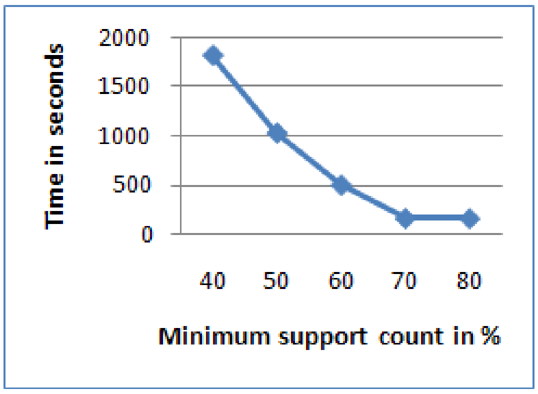

Fig 4: Real time values for CHESS data

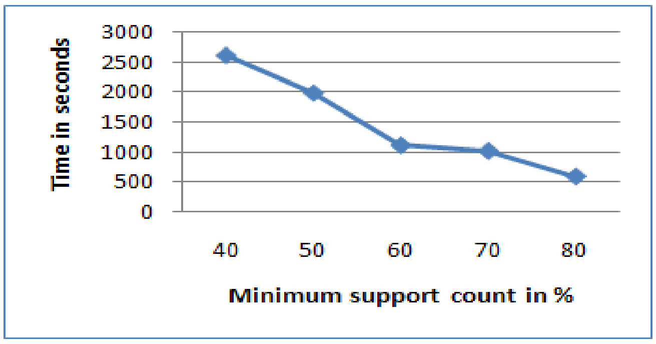

Fig 5: Real time values for T10I4D100K data

From real time observations shown above,we observed that as support count increases the time in seconds decreases.To further reduce the time and to increase the performance we need to run Apriori algorithm in parallel. For parallel implementation of Apriori algorithm on a quad core processor we are going to use OpenMP.OpenMP is an Application Programming Interface (API) which uses Fork and Join concept for parallelism [14].

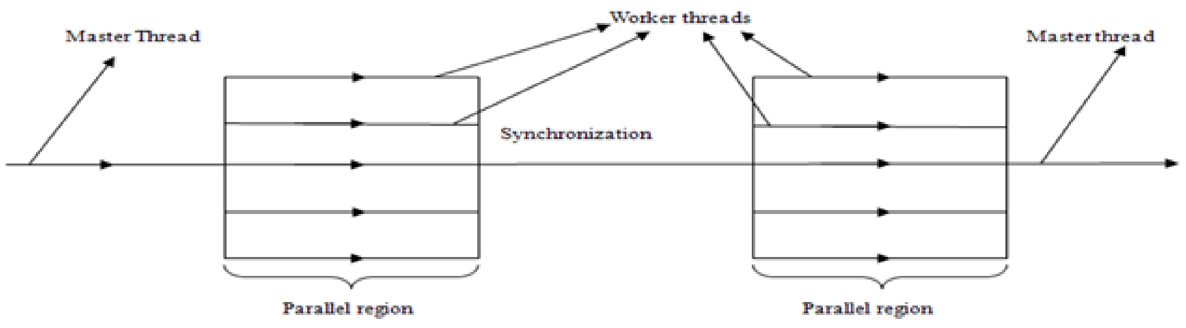

Fig 6: OpenMP Fork and Join model

In an OpenMP Fork and Join model the program begines with master thread, some part of program can be make work in parallel by creating chlid threads.Master thread work in a serial manner until parallel construct isencounterd[15].Master thread create a team of child thread which work in parallel in a parallel region.The parallel construct divides the work among the child threads equally.After the execution the master thread continues but the child threds get synchronized and enumerated[16].

V. CONCLUSION

In this paper, we are proposing a work of serial and parallel implementation of Apriori algorithm on a quad core prosessor. We have implemented Apriori algorithm on a quad core processor in a serial manner with five standard datasets using different support counts varing from 40% to 80% and Parallel implementation of Apriori algorithm on a quad core processor using OpenMP is in progress. From our proposed work we can ensure that parallel results would be better than serial once.Our aim of measuring the serial and parallel performance of Apriori agorithm on aquad core processor with respect to time and comparison between them would get satisfied.

REFERENCES

[1] Jiawei Han and Micheline Kamber, *Data Mining concepts and Techniques*2nd edition Morgan Kaufmann Publishers, San Francisco2006.
[2] Anuradha.T, Satya Pasad R and S N Tirumalarao. *Parallelizing Apriori on Dual Core using OpenMP*. *International Journal of Computer Applications* 43(24):33-39, April 2012. Published by Foundation of Computer Science, New York, USA.
[3] Chao-Tung Yang, Tzu-Chieh Chang, Hsien-Yi Wang, William C.C. Chu, Chih-Hung Chang. Performance Campion with OpenMP Parallelization for Multi-core Systems. Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications, 2011 IEEE, pp 232-237.
[4] Ying Liu, Fuxiang Gao. Parallel Implementations of Image Processing Algorithms on Multi-Core. 2010 Fourth International Conference on Genetic and Evolutionary Computing, 2010 IEEE, pp 71-74.
[5] 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
[6] Ketan D. Shah, Dr. (Mrs.) Sunita Mahajan. Performance Analysis of Parallel Apriori on Heterogeneous Nodes. 2009International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009 IEEE, pp 42-44.
[7] Anuradha.T, Satya Prasad.*Parallelizing Apriori on Hyper-Threaded Multi-Core Processor.*International Journal of Advanced Research in Computer Science and Software Engineer. *Volume 3*, *Issue 6*, June 2013.
[8] Zhong Zheng, Xuhao Chen, Zhiying Wang, Li Shen, Jiawen Li. Performance Model for OpenMP Parallelized Loops. 2011International Conference on Transportation, Mechanical, and Electrical Engineering(TMEE) December 16-18, Changchun, China, 2011 IEEE, pp 383-387.
[9] Kyung Min Lee, Tae Houn Song, Seung Hyun Yoon, Key Ho Kwon, Jae Wook Jeon. OpenMP Parallel Programming Using Dual-Core Embedded System. 2011 11thInternational Conference on Control, Automation and Systems Oct. 26-29, 2011in KINTEX, Gyeonggi-do, Korea, pp762-766.
[10] Anuradha.T, Dr.Satya Prasad.R, Dr.Tirumala Rao.S.N *Performance evaluation of apriori with memory mapped files*. International Journal of Computer Science Issues *Vol.10*, *Issue 1*, no1, January 2003.
[11] "OpenMP Application Program Interface" Version 3.0 May 2008.
[12]Tim Mattson and Larry Meadows. "A 'Hands-on' Introduction to OpenMP " Intel Corporation.
[13]Kent Milfeld. "Introduction to Programming with OpenMP" February 6th 2012, TEXAS ADVANCED COMPUTING CENTER (TACC).
[14] Multi-core Processor Wikipedia [Available] en.wikipedia.org/wiki/Multi-core processor.
[15] *Blaise Barney, Lawrence Livermore.* Introductionto Parallel Computing.https://computing.llnl.gov/tutorials/parallel_comp/.
[16] Frank Willmore. "Introduction to Parallel Computing", February 6, 2012.