IJETIE International Journal of Emerging Technology and Innovative Engineering
Volume 2, Issue 2, February 2016 (ISSN: 2394 – 6598)

Date of Publication : 31.01.2016

ENHANCING THE APRIORI ALGORITHM USING OPENMP IN ASSOCIATION RULE MINING

Mahendra D. Patale
Rajiv Gandhi College of engineering and Research, Nagpur
mahenpatle26@gmail.com

Prof. AshishGolghate
Rajiv Gandhi College of Engineering And Research, Nagpur
ashishgolghate@gmail.com

ABSTRACT

This paper contains the overview of parallelization technique to improve the performance of existing dataset in mining algorithms and make them capable of handling large amount of data. In a last few decades data in massive amount being get generated from different sections of society. To show the relation among items Association rule mining is used. Apriori Algorithm is used to show relation among frequent itemsets. For valuable knowledge data needs to be mine. Serial mining generally reduces performance and consume time. For solving serial mining problems parallel mining is introduced. In parallel mining Multi-core processors from advanced computers are used. In parallelism load is distributed among the core and in this way load balancing is done. In this paper we have implemented Apriori Algorithm in serial as well as in parallel manner using OpenMP on quad core processor and compared the results for increased performance.

General Terms
Association rule mining, Apriori algorithm.

Keywords
OpenMP, Parallelism, Multi-core processor, Apriori, Data mining.

1. INTRODUCTION

A parallel computer is a set of processors that are able to work cooperatively to solve a computational problem. This definition is broad enough to include parallel supercomputers that have hundreds or thousands of processors, networks of workstations, multiple-processor workstations, and embedded systems. Parallel computers are interesting because they offer the potential to concentrate computational resources whether processors, memory, or I/O bandwidth on important computational problems.

Association Rule mining (ARM) or frequent itemset mining is an important functionality of Data Mining (3). Data mining mainly deals with large volumes of data to extract the previously unseen and useful knowledge (1).The best algorithm for finding frequent itemsets from a transactional database is apriori algorithm. It requires scanning the entire database more number of times. The main concern should be how to improve the performance of the algorithm as data mining mainly deals with large volumes of data, one way of improving the performance of apriori is parallelizing the algorithm (6). The recent advancement in computer hardware for parallel processing is Multi-Core architectures (7). In this paper we present the performance evaluation of parallelization of apriori for different sized databases with different support counts on a quad core system compared to its sequential implementation using an efficient and easy technique with perfect load balancing between the processors.

.

2. RELATED WORK

Ketan shah and SunitaMahajan 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 SatyaPrasad.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]. RakeshAgrawal and Ramakrishna srikant presents two new algorithms Apriori and AprriroriTid and combined into an Apriori hybrid algorithm and demonstrate in scale-up-properties [5]. Kyung Min Lee, Tae Houn Song evaluates a parallel programming model, parallel programs and OpenMP 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].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]. Anuradha.T, Dr.SatyaPrasad.R, Dr.TirumalaRao.S.N, gives the performance of Apriori using Linux mmap() function compared to fread() function in both the serial and parallel environments [10]. Jaiwen Li, Zhang Zheng, Xuhoo Chen, Li Shen, and Zhiying Wang give a performance model for OpenMP parallelized loops to address the critical factors which influences the performance [8]. Ying Liu and FuxiangGao present the cubic convolution interpolation algorithm for image processing and parallelized it by using OpenMP on multi-core processors [4].

3. THEORETICAL BACKGROUND

3.1 Multi-core processor

Multi core means two or more processors. But they differ from independent parallel processors as they are integrated on the same chip circuit. A multi core processor implement message passing or shared memory inters core communication methods for multiprocessing [7]. If the number of threads are less than or equal to the number of cores, separate core is allocated to each thread and threads run independently on multiple cores. If the number of threads is more than the number of cores, the cores are shared among the threads. Any application can be threaded can be mapped efficiently to multi-core processor, the improvement in performance gained by the use of multi core processors depends on the fraction of the program that can be parallelized [14]

3.2 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: To find the count of each candidate in Ck the entire database is scanned. Candidate k-itemset is represented by Ck. To find whether that itemset can be placed in frequent k-itemsetLk to count each itemset in Ck is compared with a predefined minimum support count [10].

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. If the database size is large, so to find all the frequent itemsets in the database, it requires more time [10].

3.3 OpenMP

OpenMP supports multi-platform shared memory multiprocessing programming in C, C++ and FORTRAN on much architecture which including UNIX and Microsoft Windows platforms [2]. It consists of a set of variables that influence run-time behavior such as library routines, compiler directives [12].

3.4 Title and Authors

The title (Times New Roman 18-point bold), authors' names (Times New Roman 12-point (BOLD)) and affiliations (Times New Roman 12-point) run across the full width of the page – one column wide. We also recommend e-mail address (Times New Roman 12-point). See the top of this page for three addresses. If only one address is needed, center all address text. For two addresses, use two centered tabs, and so on. For three authors, you may have to improvise.

4. EXPERIMENTAL WORK

We are going to enhance an existing Apriori algorithm in such a way that the execution time to execute an Apriori algorithm with a given data set will take minimum time when we execute it with the parallel execution processing with the help of OpenMp tool using an association rule miming as compaired to the serial execution of those datasets.

4.1 OVERVIEW OF PLAN OF ACTION

AIM :To Parallelize Apriori algorithm using MPI for better performance.

OBJECTIVE :

i) Implementation of Apriori algorithm in a Serial manner.

ii) Implementation of Apriori algorithm in a Parallel manner using MPI.

iii) Comparing Apriori algorithm's Serial result and Parallel result for better performance with respect to time.

5. ACKNOWLEDGMENTS

Here in, I would like to appreciate Prof. AshishGolghate for their great support and encouragement during this research. Thank you, dear professors..

6. REFERENCES

[1] Han and MichelineKamber, *Data Mining concepts and Techniques* 2nd edition Morgan Kaufmann Publishers, San Francisco 2006.

[2] Anuradha.T, Satya Prasad 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, FuxiangGao.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.) SunitaMahajan. Performance Analysis of Parallel Apriori on Heterogeneous Nodes.2009 International 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] ZhongZheng, Xuhao Chen, Zhiying Wang, Li Shen, Jiawen Li. Performance Model for OpenMP Parallelized Loops. 2011 International 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 WookJeon.OpenMP Parallel Programming Using Dual-Core Embedded System. 2011 11th International Conference on Control, Automation and Systems Oct. 26-29, 2011 in KINTEX, Gyeonggi-do, Korea, pp762-766.

[10] Anuradha.T, Dr.SatyaPrasad.R, Dr.TirumalaRao.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.*Introduction to ParallelComputing.https://computing.llnl.gov/tutorials/parallel_comp/