Parallel Computing 40 (2014) 768–785

Contents lists available at ScienceDirect

Parallel Computing

journal homepage: www.elsevier.com/locate/parco

Novel parallel method for association rule mining on multi-core shared memory systems

Lan Vu *, Gita Alaghband

Dept. of Computer Science and Engineering, University of Colorado Denver, Denver, CO 80204, USA

ARTICLE INFO

Article history:
Available online 11 October 2014

Keywords:
Frequent pattern mining
Multi-core
Shared memory
Association rule mining
Parallel algorithm
Databases

ABSTRACT

Association rule mining (ARM) is an important task in data mining with many practical applications. Current methods for association rule mining have shown unstable performance for different database types and under-utilize the benefits of multi-core shared memory machines. In this paper, we address these issues by presenting a novel parallel method for finding frequent patterns, the most computational intensive phase of ARM. Our proposed method, named ShaFEM, combines two mining strategies and applies the most appropriate one to each data subset of the database to efficiently adapt to the data characteristics and run fast on both sparse and dense databases. In addition, our new lock-free design minimizes the synchronization needs and maximizes the data independence to enhance the scalability. The new structure lends itself well to dynamic job scheduling resulting in a well-balanced load on the new multi-core shared memory architectures. We have evaluated ShaFEM on 12-core multi-socket servers and found that our method run up to 5.8 times faster and consumes memory up to 7.1 times less than the state-of-the-art parallel method. For some test cases, ShaFEM can save up to 4.9 days of execution time over the compared method.

1. Introduction

Association rule mining (ARM) is one of the fundamental tasks in data mining. Since its first application for the analysis of sales or basket data which was introduced by Agrawal et al. [1], ARM has been applied broadly in many fields with an increasing number of applications such as market analysis, biomedical and computational biology research, web mining, decision support, telecommunications alarm diagnosis and prediction, and network intrusion detection [2,4,7,8,13,14,46]. Because of the importance of this mining task, ARM has become an essential mining component of most popular database systems like Oracle Database (RDBMS), Microsoft SQL Server, IBM DBS2 Database and IBM DBS2 and statistical software like R, SAS and SPSS Clementine [24–26,43–45]. The increasing need to analyze big data has led to the development of new ARM method that can leverage the computing power of emerging platforms to support this mining task. Furthermore, widening the applicable areas of ARM requires algorithms that can perform efficiently on different data types.

1.1. Motivation

Several studies have shown that ARM methods typically worked well for certain types of databases. Most methods performed efficiently on either sparse or dense databases but poorly on the other [11,15,17–22,28,30]. Table 1 presents

* Corresponding author.
E-mail addresses: Lan.Vu@ucdenver.edu (L. Vu), Gita.Alaghband@ucdenver.edu (G. Alaghband).

http://dx.doi.org/10.1016/j.parco.2014.08.003
0167-8191/

Table 1
Running time on sparse and dense database.

Databases	Type	*Minsup* (%)	Apriori	Eclat	FP-growth
Chess	Dense	20	1924	77	89
Connect	Dense	30	522	366	403
Retail	Sparse	0.003	18	59	10
Kosarak	Sparse	0.08	4332	385	144

the execution time of three well-known sequential algorithms Apriori [1], Eclat [10] and FP-growth [11] on sparse and dense databases. It shows Eclat performs best on dense data while FP-growth runs fastest on the sparse ones (underline numbers indicate the best execution times among the three algorithms).

Furthermore, the large data size and the amount of computation involved lead to the crucial need of applying parallel computing for this mining task to speed up the large-scale data mining application. Most existing works have proposed parallel solutions for distributed-memory systems [9,34,35,37,38,40]. Some surveys [34,35] show that very few studies were conducted on parallel frequent pattern mining algorithms for the shared memory multi-core platforms. Most of them have based on Apriori that is far less efficient than the other algorithms (shown in Table 1). None of previous parallel work took into consideration the data characteristics to improve the mining performance on different database types.

1.2. Contributions

We present a novel parallel ARM method named ShaFEM for the new multi-core shared memory platforms to solve the above issues. The proposed method uses a new data structure named XFP-tree that is shared among processes to compact data in memory. Then, each parallel process independently mines rules and based on the density of mining data being processed dynamically selects and switches between two mining strategies where one is suitable for sparse data and the other works well on dense data. The main contributions of our study include:

(1) A novel parallel mining method that can dynamically switch between its two mining strategies to adapt to the characteristics of the database and run fast on both sparse and dense databases. This original contribution is based on the recognition for the need to apply different data mining strategies as mining proceeds and the fact that the dataset characteristics change during this processing, and therefore the need for runtime detection of when this should occur.
(2) A new efficient parallel lock free approach that applies new data structures to enhance the independence of parallel processes, minimize the synchronization cost and improve the cache utilization. Additionally, its dynamic job scheduling for load balancing helps increase the scalability on multi-core shared memory systems. This is an important contribution as ARM is a challenging problem for high performance computing. It has many dependent subtasks, unpredictable workload and complex data structures and requires many reduction steps.
(3) We demonstrate the efficiency of our approach by conducting intensive experiments to benchmark ShaFEM and other state-of-the arts mining approaches. We present an in-depth analysis of the impact of each technique employed and the contributions made to the final performance of ShaFEM.

1.3. Paper organization

The rest of the paper is organized as follows. Section 2 introduces the problem statement and related works. The parallel frequent pattern mining algorithm, ShaFEM, is presented in Section 3. The first mining stage to construct the XFP-tree is demonstrated in Section 4. Section 5 details the second mining stage and describes the dynamic decision making process to switch between the two mining strategies. We evaluate the scalability and analyze the performance merits of ShaFEM in Section 6. The final section is our conclusion.

2. Background

2.1. The problem statement

Association rule mining (ARM) aims at discovering rules that specify the frequency co-occurrence of groups of itemsets, subsequences, or substructures in a database. For example, an association rule of retail database can be of the form "70% of customers who buy milk and butter also buy bread with confidence 90%". Detection of these interesting rules contributes to the knowledge base used to build intelligence systems such as product recommendation, gene function prediction, network intrusion detection, search engine ranking, etc. Google uses this mining task for their query recommendation system [9].

The association rule mining problem can be stated as follows: Let $I = \{i_1, i_2, \ldots, i_n\}$ be the set of n distinct items in the transactional database D. Each transaction T in D contains a set of items called *itemset*; a *k-itemset* is an itemset with k items. The *count* of an *itemset* x is the number of occurrences of x in D and the *support* of x is the percentage of transactions containing x.