Data Reordering for Minimizing Threads Divergence in GPU-Based Evaluating Association Rules

Youcef Djenouri[1], Ahcene Bendjoudi[1], Malika Mehdi[2], Zineb Habbas[3], and Nadia Nouali-Taboudjemat[1]

[1] DTISI, CERIST Research center, rue des freres Aissou, Benaknoun, Algiers, Algeria
[2] MoVeP Lab, USTHB University, Algiers, Algeria
[3] LCOMS, University of Lorraine Ile du Saulcy, 57045 Metz Cedex, France
{y.djenouri,malika.mehdi}@gmail.com, {abendjoudi,nnouali}@cerist.dz, zineb@univ-lorraine.fr

Abstract. This last decade, the success of Graphics Processor Units (GPUs) has led researchers to launch a lot of works on solving large complex problems by using these cheap and powerful architecture. Association Rules Mining (ARM) is one of these hard problems requiring a lot of computational resources. Due to the exponential increase of data bases size, existing algorithms for ARM problem become more and more inefficient.Thus, research has been focusing on parallelizing these algorithms. Recently, GPUs are starting to be used to this task. However, their major drawback is the threads divergence problem. To deal with this issue, we propose in this paper an intelligent strategy called Transactions-based Reordering "TR" allowing an efficient evaluation of association rules on GPU by minimizing threads divergence. This strategy is based on data base re-organization. To validate our proposition, theoretical and experimental studies have been carried out using well-known synthetic data sets. The results are very promising in terms of minimizing the number of threads divergence.

Keywords: Association Rules Mining, GPU Computing, Threads Divergence.

1 Introduction

Finding hidden patterns from a large data base is a challenging problem in articficial inteligence community (AI). Association Rules Mining (ARM) is one of the most used techniques allowing the extraction of useful rules among a transactional data base. Because of the exponential increasing of data bases size, the existing ARM algorithms are highly time consuming. Parallel computing is then used to reduce the computation time of highly complex ARM algorithms. Recently, Graphics Processor Units (GPUs) are considered as one of the most used parallel hardware to solve large scientific complex problems. Indeed, nowaday computer architectures are hybrid. Hybrid architectures are mainly composed of

S. Omatu et al. (eds.), *Distributed Computing & Artificial Intelligence, 12th Int. Conference*, Advances in Intelligent Systems and Computing 373, DOI: 10.1007/978-3-319-19638-1_6

two components, a CPU containing a multi-core processor and a main memory and the GPU host composed of hundreds of computing cores (threads) and a hierarchy of memories with different carachteristics. The GPU threads are grouped on blocs and a bloc is scheduled on several warps containg generally 32 threads executed in parallel. Threads within the same warp follow the SIMD model (Single Instruction Multiple Data) and should execute the same instruction at any time. This property can be the source of threads divergence when distinct threads of the same warp are constrained to execute different instructions.

Several GPU-based ARM algorithms have been proposed in the literature ([1-3, 6 and 8]). These algorithms reduce perfectly the execution time of the ARM process. However their major drawback is the threads divergence problem. In association rules mining, the rule evaluation is time consuming. The evaluation of the rules consists on calculating two statistical measures (the support and the confidence of each rule). To determine these measures, transactional data bases must be entirely scanned for each evaluation. In [8], the authors proposed an efficient threads mapping strategy to evaluate the rules. Each thread is mapped with one transaction and a single rule. After that, sum reduction is used to compute the global evaluation of such rule. This technique induces threads divergence between threads because the transactions have in general different data values with different sizes (see Section 3 for more details).

The main goal of this paper is to propose an efficient strategy minimizing GPU-threads divergence in the rules evaluation process. The idea is to perform a pre-processing step before the mining process. The transactional data base is re-organized according to the length of each transaction. This operation allows grouping the set of transactions having the same size and then assign them to the same warp. However, it causes unload balancing between GPU blocs. To deal with this issue, we propose a new strategy to be added to the pre-processing step. The transactions are first sorted according to the number of items, then assigned successively to the threads. To validate the proposed approaches, several experiments have been carried out on well-known synthetic data sets. The results are very promising in terms of the number of threads divergence and then improving the overall execution time.

The remainder of the paper is as follows, the next section presents a state-of-the-art on GPU-based ARM algorithms. Section 3 explains the GPU-threads divergence in the evaluation rules process. Section 4 presents the proposed strategy to avoid threads divergence. Analytical and experimental study of the suggested strategy are detailed in Section 5. Finally, we conclude this work in Section 6.

2 Related Work

Association rule mining attempts to discover a set of rules covering a large percentage of data and tends to produce an important number of rules. However, since the data bases are increasingly large, ARM process has became a high intensive computing.