Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 369 – 378

Procedia Computer Science

International Conference on Computational Science, ICCS 2013

G-DBSCAN: A GPU Accelerated Algorithm for Density-based Clustering

Guilherme Andrade[a], Gabriel Ramos[a], Daniel Madeira[a,b], Rafael Sachetto[a], Renato Ferreira[c], Leonardo Rocha[a,*]

[a]*Federal University of São João del-Rei, MG, Brasil*
Computer Science Departament
[b]*Fluminense Federal University, RJ, Brasil*
Computer Science Institute
[c]*Federal University of Minas Gerais, MG, Brasil*
Computer Science Departament

Abstract

With the advent of Web 2.0, we see a new and differentiated scenario: there is more data than that can be effectively analyzed. Organizing this data has become one of the biggest problems in Computer Science. Many algorithms have been proposed for this purpose, highlighting those related to the Data Mining area, specifically the clustering algorithms. However, these algorithms are still a computational challenge because of the volume of data that needs to be processed. We found in the literature some proposals to make these algorithms feasible, and, recently, those related to parallelization on graphics processing units (GPUs) have presented good results. In this work we present the G-DBSCAN, a GPU parallel version of one of the most widely used clustering algorithms, the DBSCAN. Although there are other parallel versions of this algorithm, our technique distinguishes itself by the simplicity with which the data are indexed, using graphs, allowing various parallelization opportunities to be explored. In our evaluation we show that the G-DBSCAN using GPU, can be over 100x faster than its sequential version using CPU.

Keywords: clustering, dbscan, parallel computing, GPU

1. Introduction

With the advent of Web 2.0, we observed a real democratization of the data generation. Several tools are being developed allowing anyone with Internet access to publish and distribute data with a speed never seen before. The large volume of data generated, as well as the high complexity of its relations, has generated in recent years a challenging scenario for several applications: there is more data than that can be effectively analyzed. Thus organizing and finding appropriate information resources to fulfill the needs of users has become one of the most challenging problems in computer science.

New proposals for models and algorithms that are able to handle this data efficiently (response time and appropriate use of computational resources) and effectively (response quality, or the robustness and accuracy to perform a task) are emerging every moment. Among these, we highlight those related to the Data Mining area [1]. Data

*Corresponding author. email: lcrocha@ufsj.edu.br ; Tel.: +55-032-3373-3985 .

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science
doi:10.1016/j.procs.2013.05.200

mining applications are highly relevant because of their wide applicability in terms of tasks and target scenarios, improving the quality and variety of the functionalities provided by all sorts of information systems. Nevertheless, they are still a computational challenge because of the data volume to be processed and the irregular nature of most of the existing algorithms, which make both their performance and resource demands quite unpredictable.

A collection of algorithms that illustrates this scenario are those related with clustering [2]. The goal of these algorithms is to organize large sets of objects into different groups (clusters) according to a similarity metric. These techniques can be used in many different scenarios, such as social networks, recommendation systems, bioinformatics etc., making their use even more challenging. In the literature, strategies has been proposed to make these applications feasible, whether through data indexing techniques [3, 4], either through the parallelization of these tasks using different processing units [5, 6]. With respect to parallelization strategies, the use of graphics processing units (GPU's) [7] has been given considerable importance, since these are able of providing a higher level of parallelism than multicore CPU's, associated with a lower energy consumption [8].

Thus, in this work we present a new clustering algorithm, the *G-DBSCAN*, a GPU accelerated algorithm for density-based clustering. Our algorithm is based on the original DBSCAN proposal [9], one of most important clustering techniques, which stands out for its ability to define clusters of arbitrary shape as well as the robustness with which it deals with the presence of data noise. The implementation strategy is quite simple and is divided into two steps. The first step is to construct a graph that will represent the data, where each object is represented as a node in the graph and an edge is created between two objects when the similarity measure between them is less than or equal to a threshold defined as an input parameter (i.e. Euclidean distance less than 3). After the construction of this graph, the second step is to identify the clusters, using a traditional breadth-first search (BFS) to traverse the graph created in the first step. In this work, both steps were implemented using GPUs, resulting in an extremely efficient algorithm regarding to the execution time, achieving a speedup greater than 100×.

The remainder of this paper is divided as follows: in Section 2 we describe some related work. In Section 3 we present the original DBSCAN proposal, detail the implementation strategy based on graphs used in the G-DBSCAN as well as the parallelization strategies. Section 4 presents the experiments performed to evaluate our algorithm, and the obtained results. Finally, we present our conclusions in Section 5.

2. Related Work

Data clustering is one of the most common and more used techniques in data mining. Its goal is basically, receiving a dataset as input, organize the data into semantically consistent groups, based on a previously defined similarity metric. In [2] several issues related to the use of clustering techniques are presented, highlighting some of its challenges, such as how to properly set the input parameters, how to specify a good similarity measure metric and how to work with large volumes of data.

Several clustering algorithms are found in the literature [10]. These algorithms range from simpler techniques and widely used in various scenarios, such as k-means [11] to more elaborate and context-driven techniques, such as subspace clustering [12] and partitioning clustering [13]. A set of clustering techniques which is receiving great attention is the one related to density-based clustering [9, 14, 15]. Such techniques are distinguished by their ease of implementation and by the applicability in different contexts. Moreover, these techniques do not need to determine in advance, as an algorithm input, the number of clusters, as is done by the others techniques mentioned above.

Among the density-based clustering techniques aforementioned, the most referenced in the literature is DBSCAN [9]. The DBSCAN technique is even being used as a base for many other techniques [14, 5]. Its operation is based on calculating a proximity radius between each pair of objects, which is defined according to the adopted similarity metric (i.e. Euclidean distance, cosine similarity, etc.). From a minimum proximity radius, defined as an algorithm input, objects are grouped with each other whenever they are within this proximity radius. One of the most used strategies to improve the performance of these algorithms is the data indexing [3, 4]. Among the most commonly used indexing techniques, we can highlight the priority R-Tree [16], that reduces the complexity of the DBSCAN algorithm from $O(n^2)$ to $O(nlogn)$.

Although the use of data indexing techniques improves the performance of density-based clustering algorithms [5], the scalability of these algorithms and making them effectively applicable in a large data volume