Neurocomputing 169 (2015) 158–168

Contents lists available at ScienceDirect

Neurocomputing

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

Formalizing computational intensity of big traffic data understanding and analysis for parallel computing

Yingjie Xia [a,b,c,*], Jinlong Chen [b], Chunhui Wang [b]

[a] College of Computer Science, Zhejiang University, 310012 Hangzhou, China
[b] Hangzhou Institute of Service Engineering, Hangzhou Normal University, 310012 Hangzhou, China
[c] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, 215006 Soochow, China

ARTICLE INFO

Article history:
Received 21 March 2014
Received in revised form
5 October 2014
Accepted 5 October 2014
Available online 16 May 2015

Keywords:
Intelligent transportation systems
Computational intensity
Computational domain
Formalization
Parallel computing

ABSTRACT

Nowadays, traffic data can be collected from different types of sensors widely-deployed in urban districts. Big traffic data understanding and analysis in intelligent transportation systems (ITS) turns out to be an urgent requirement. This requirement leads to the computation-intensive and data-intensive problems in ITS, which can be innovatively resolved by using Cyber-Infrastructure (CI). A generic process for the solution contains four steps: (1) formalized data understanding and representation, (2) computational intensity transformation, (3) computing tasks creation, and (4) CI resources allocation. In this paper, we firstly propose a computational domain theory to formally represent heterogeneous big traffic data based on the data understanding, and then use data-centric and operation-centric transformation functions to evaluate the computational intensity of traffic data analysis in different aspects. Afterwards, the computational intensity is leveraged to decompose the domain into sub-domains by octree structure. All the sub-domains create computing tasks which are scheduled to CI resources for parallel computing. Based on the evaluation of overall computational intensity, an example of fusing Sydney Coordinated Adaptive Traffic System (SCATS) data and Global Positioning System (GPS) data for traffic state estimation is parallelized and executed on CI resources to test the accuracy of domain decomposition and the efficiency of parallelized implementation. The experimental results show that the ITS computational domain is decomposed into load-balanced sub-domains, therefore facilitating significant acceleration for parallelized big traffic data fusion.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The quantity of traffic data collected from different types of sensors widely deployed in urban districts have increased dramatically in the recent few decades, with the data quality improving significantly. This trend will continue in the foreseeable future, and lead to the urgent requirement of big traffic data understanding and analysis in intelligent transportation systems (ITS) [1]. A solution involves the use of Cyber-Infrastructure (CI) which can support the analysis of computation-intensive and data-intensive problems in high performance [2]. This solution generically consists of four steps: (1) formalized understanding and representation of heterogeneous big traffic data; (2) computational intensity evaluation on ITS applications; (3) algorithmic parallelization to create computing tasks; and (4) CI scheduling and allocation for parallel computing. For each step, the output of its previous step is used as its input. Therefore, in order to efficiently parallelize the data-driven ITS applications in step (3), we need to formalize the computational intensity of big traffic data analysis based on the formalized data understanding and representation in steps (1) and (2).

Traditional computational complexity theory is used to assess the computational intensity of traffic data analysis on the algorithmic complexity notation [3]. However, this notation merely focuses on the evaluation of algorithmic structure, and does not adequately capture spatio-temporal characteristics of data and operations for analysis. These characteristics are always dependent on spatio-temporal clustering, neighborhood, autocorrelation, and interaction dynamics of big traffic data, and their transformed computational intensity can be measured in different aspects.

Following the aforementioned solution steps and characteristics of computational intensity, in this paper we propose the architecture of our work as the flowchart in Fig. 1. We firstly define an ITS computational domain to represent multi-sensor heterogeneous data in an accommodating structure. The computational domain is defined as a high-dimensional data space consisting of a large amount of cell tuples, whose attributes include spatio-temporal information and traffic features. Based

* Corresponding author at: College of Computer Science, Zhejiang University, 310012 Hangzhou, China. Tel.: +86 571 28866898; fax: +86 571 28866717.
E-mail address: xiayingjie@zju.edu.cn (Y. Xia).

http://dx.doi.org/10.1016/j.neucom.2014.10.104
0925-2312/© 2015 Elsevier B.V. All rights reserved.

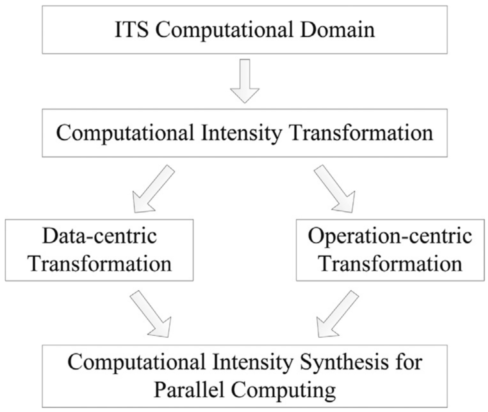

Fig. 1. Architecture of our work.

on the domain, two types of computational intensity transformation functions, data-centric function and operation-centric function, are adopted to elucidate the computational intensity of a particular big traffic data analysis in three principle aspects, memory, I/O, and computing time. Different aspects of computational intensity are synthesized to decompose the computational domain into load-balanced sub-domains, which create computing tasks to be executed in parallel. CI resources are allocated to all tasks according to the matching of computing capability and computational intensity, to improve the efficiency of the parallel computing. The formalization of computational intensity provides a comprehensive evaluation of algorithmic complexity and correlation between neighboring data cells, and therefore can facilitate to accelerate big traffic data analysis because of the accurate domain decomposition by overall computational intensity.

The rest of the paper is organized as follows. Some related work about data-driven ITS and computational complexity analysis is reviewed in Section 2. Section 3 introduces the ITS computational domain theory to formally represent multi-sensor heterogeneous traffic data. Section 4 specifies the data-centric and operation-centric transformation functions to implement the computational intensity transformation into different aspects. We also define the overall computational intensity in Section 4. An example application of data-driven ITS, multi-sensor data fusion for traffic state estimation, is illustrated in Section 5. Section6 analyzes the utilization of computational intensity for parallel computing by using real traffic data, and conducts load-balance, accuracy, and efficiency tests on the data. Finally, the conclusion with remarks on future work is drawn in Section 7.

2. Related work

Recently, the conventional ITS is evolving into the data-driven ITS where data collected from multiple traffic sensors play an essential role in ITS. Based on the types of traffic sensors used, the way to process data, and the specific applications, a full data-driven ITS can be classified. The main classified categories include vision-driven ITS, multisource-driven ITS, learning-driven ITS, and visualization-driven ITS [1].

Vision-driven ITS takes the traffic data collected from video sensors as input, and uses the processing output for ITS related applications, such as (1) traffic object detection [4], monitoring [5], and recognition [6], (2) traffic behavior analysis [7], (3) traffic data statistical analysis [4], and (4) vehicle trajectory construction [8]. However, vision-driven ITS suffers from the environmental constraints, e.g., snow, static or dynamic shadows, and rain [9]. Therefore, multisource-driven ITS uses multiple types of sensors, such as Sydney Coordinated Adaptive Traffic System (SCATS) loop detector, Global Positioning System (GPS), and Remote Traffic Microwave Sensor (RTMS), to play complementary roles for each other. The collected heterogeneous traffic data can be fused for traffic state estimation [10], clustered for urban advertising value evaluation [11], and combined to improve the reliability of vehicle classification [12]. Although video devices and multiple sensors can generate traffic data for different applications in ITS, they still require some learning tools, including online learning [13], rough set theory [14], adaptive dynamic programming (ADP) [15], and fuzzy logic [16], to extract intrinsic mechanisms from historical and real-time data in specific applications. This is named learning-driven ITS. The output of data processing by learning tools can be visualized to help people understand and analyze traffic data intuitively in visualization-driven ITS [17]. Some visualization packages, such as CubeView, are developed to identify abnormal traffic patterns and accordingly take the system back to the normal track [18].

Although all the aforementioned work is related to the data-driven ITS, to the best of our knowledge, little research has been carried out on how to deal with the data growing in a large scale. For example, Google cooperates with INRIX to use the GPS data collected from more than 30,000,000 taxies, transit vehicles, and trucks, to estimate the traffic states plotted on the Google map [19]. These massive GPS data are processed by high performance computers (HPC), which divide the data by cities and create the corresponding multiple computing tasks. The performance of this implementation is limited by the unbalanced computing loads of different cities, and can be improved by achieving load-balance. As the solution, the goal of this paper is to create load-balanced computing tasks based on the evaluation of computational complexity. The computational complexity theory dates from 1960s, and is firstly used to evaluate the polynomial time on Turing machines [20]. This topic came into the picture owing to the discovery of NP-complete problems in 1970s. The NP-completeness can indicate the computational complexity of problems by using enumerative optimization methods and approximation algorithms [21]. Loosely speaking, an algorithm can be described as a finite sequence of instructions, and its computability can be quantified as the computational complexity measured by how fast a computer works out all instructions [22]. The analysis of computational complexity leads to various postprocessing models, such as parallel computing, probability calculation, and decision tree. For example, the evaluated complexity of rules extraction from massive traffic data is used to parallelize the attribute significance calculation in bootstrapping rough set algorithm to estimate traffic state more accurately and efficiently [23]. The work in [23] focuses on the evaluation of algorithmic complexity for the algorithmic parallelization of computation-intensive problems. However, massive traffic data analysis urgently requires the evaluation of data-centric computational complexity to parallelize the data-intensive problems. It is also argued that the computational complexity is different from the computational intensity. The computational complexity theory addresses how much the intrinsic complexity of computing tasks are, while the computational intensity requires much extra consideration on the spatio-temporal correlated characteristics in massive traffic data and algorithms. In geographical information systems (GIS), the computational intensity of spatial analysis has been proposed to discover the difficult nature of spatial domain decomposition. This work is fundamental to analyze computation-intensive spatial problems which focus on the geographic data in two-dimensional space [24]. However, in current ITS research, few efforts have been made to formalize the computational intensity with respect to the traffic data characteristics, quantity, and spatio-temporal distribution. Therefore, this is what we aim to elucidate in the following parts.

Download English Version:

https://daneshyari.com/en/article/6865561

Download Persian Version:

https://daneshyari.com/article/6865561

Daneshyari.com