Microprocessors and Microsystems 000 (2017) 1–8

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

An integrated architecture for future studies in data processing for smart cities

Cristian Chilipirea, Andreea-Cristina Petre, Loredana-Marsilia Groza, Ciprian Dobre, Florin Pop*

Computer Science Department, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Romania

ARTICLE INFO

Article history:
Received 12 February 2016
Revised 24 January 2017
Accepted 6 March 2017
Available online xxx

Keywords:
Architecture
Big data
Data processing
Crowd sensing
Crowd dynamics
Intelligent transportation systems

ABSTRACT

Data processing for Smart Cities become more challenging, facing with different handling steps: data collection from different heterogeneous sources, processing sometimes in real-time and then delivered to high level services or applications used in Smart Cities. Applications used for intelligent transportation systems, crowd management, water resources management, noise and air pollution management, require different data processing techniques. The main subject of this paper is to propose an architecture for data processing in Smart Cities. The architecture is oriented on the flow of data from the source to the end user. We describe seven steps of data processing: collection of data from heterogeneous sources, data normalization, data brokering, data storage, data analysis, data visualization and decision support systems. We consider two case studies on crowd management in smart cities and on Intelligent Transportation Systems (ITS) and we provide experimental highlights.

1. Introduction

More and more applications today use, generate and handle very large volumes of data. In particular, this is true for Smart City applications, which attract a rapidly increasing interest from government, companies, citizens, developers, scientists, etc. They cover a large spectrum of needs in public safety, water and energy management, smart buildings, government and agency administration, social programs, transportation, health, education. They are fed with huge amounts of input data, in various formats, from a continuously increasing number of sources (sensors, governmental, regional, and municipal sources, citizens, public open data sources, etc.), and are described by a complex workflow and in many cases impose real-time processing capabilities, useful in decision taking.

The large volume of data coming from a variety of sources and in various formats, with different storage, transformation, delivery or archiving requirements, complicates the task of context data management. At the same time, fast responses are needed for real-time applications. Despite the potential improvements of the Smart City infrastructure, the number of concurrent applications needing quick data access will remain very high. With the emergence of the recent cloud infrastructures, achieving highly scalable data management in such contexts is a critical challenge, as the overall application performance is highly dependent on the properties of the data management service.

Extracting valuable information from raw data is especially difficult considering the velocity of growing data from year to year and the fact that 80% of data is unstructured. In addition, data sources are heterogeneous (various sensors, users with different profiles, etc.) and are located in different situations or contexts. This is why the Smart City infrastructure runs reliably and permanently to provide the context as a “public utility” to different services. Context-aware applications exploit the context to adapt accordingly the timing, quality and functionality of their services. The value of these applications and their supporting infrastructure lies in end-users always operating in a context: their role, intentions, locations and working environment constantly change.

As the scale, complexity and dynamism of distributed systems is dramatically growing, their configuration and data management have started to become a limiting factor of their development. This is particularly true in the case of Cloud is used for data storage and also for data processing, where the task of managing hundreds or thousands of nodes while delivering highly reliable services entails an intrinsic complexity. Furthermore, Cloud computing introduces another challenge which impacts on the resource management decisions. In these contexts, self-management mechanisms have to take into account the cost-effectiveness of the adopted decisions.

Considering all of these aspects, the main subject of this paper is to propose an architecture for Big Data processing in Smart

* Corresponding author.
E-mail address: florin.pop@cs.pub.ro (F. Pop).

http://dx.doi.org/10.1016/j.micpro.2017.03.004
0141-9331/

Cities. The architecture is oriented on the flow of data from the source to the end user. We describe seven steps of data processing: collection of data from heterogeneous sources, data normalization, data brokering, data storage, data analysis, data visualization and decision support systems. We describe two case studies on crowds' management in smart cities and on Intelligent Transportation Systems (ITS).

The paper is structured as follows. Section 2 presents the related work on crowd data smart cities and on ITS. The proposed architecture is presented in Section 3. Two use cases are described in Section 4. Then, the experiments obtained for these use cases are presented in Section 5. The paper ends with conclusions and future work presented in Section 6.

2. Related work

Smart Cities [1] represent an important goal which can dramatically improve the life of citizens. There is a lot of research aiming to get us closer and closer to this goal. The idea of a smart city is in accordance to other movements in research such as Internet of Things [2–4] and Big Data [5]. New York times actually declared this period the "Age of Big Data".[1]

In order to enable Smart Cities technologies such as Internet of Things, Wireless Sensor Networks [6] and Crowd Sensing [7] are the catalysts providing data about our cities. The need for sensing in Smart Cities is explored in [8]. This data needs to be processed often using Big Data techniques in order to extract the information required to make decisions about the cities. This information and the decisions are then used in order to inform the citizens to take certain actions or to activate actuators for enabling automatic processes. A good example where actuators can improve Smart Cities is given by the management of green spaces [9].

Probably the most important issues addressed in order to build Smart Cities are the ones of Crowd Dynamics [10]. In order to understand Crowd Dynamics, we need data on the movements of as many people as possible. These movements need to be recorded for both pedestrians [11] and for vehicles [12]. The problem of tracking is not solved in any of the two scenarios. This is surprising, considering the problem of tracking a particular individual is usually solved by the use of GPS [13]. However, GPS requires user participation which is difficult to obtain, in contrast WiFi [14] or cellular methods [15] can be used to gather data on large crowds. These systems also do not work indoors and require the cooperation of the individual being tracked in order to generate a position estimate.

It is important to treat both indoor and outdoor cases when considering human mobility. This is because modern vital facilities, such as hospitals, which are part of the backbone of many cities consist of large areas with multiple buildings. An example of how the dynamics inside these facilities can be used in order to improve the layout of the facility is given by Ruiz-Ruiz et al. [16]. Similarly, Universities campuses, another type of large facilities at the core of cities, are analyzed [17,18] in order to better understand the dynamics inside them.

Crowd tracking experiments are taking place in a wider variety of places like mass events [19] or festivals [20]. They are also used in order to measure queues using only WiFi signals [21]. This queue can represent waiting time at a counter, which directly affects customer experience or the movement through security lines at an airport [22].

Crowd Sensing can be used in order to extract all types of data for smart cities. A powerful example is given by Han et al. [23] where students are asked to take pictures of plants around the campus. The pictures are then analyzed by scientists in order to better understand the status of flora. Projects like this could potentially be used at the scale of a city in order to measure a large variety of features. It is not always necessary for people to be active in their participation of data gathering. Passive systems require only their presence in the monitored location, which can even be obtained in an opportunistic manner. Whenever any citizens carrying the scanner walks or drives on a specific street data about the street can be gathered. In this way maps can be enhanced with features [24] such as roundabouts or pot-holes. Diverse uses include even earthquake detection [25] and soon maybe even the detection of effects produced by these large natural disasters.

There are many projects and platforms targeted directly at crowd sensing: Medusa [26], Matador [27], Mosden [28] and mCrowd [29]. And these platforms already implement important features for Smart Cities such as crowd sources new reporting [30] but they do not yet combine the data sets or offer a method to analyze the data in order to extract information hidden inside it. This type of information represents answers to questions that we don't yet have and they can currently only be obtained by using Big Data techniques.

The data gathered from all these systems is usually analyzed by experts or scientist manually. This is the case for [16], where categorization of individuals into different groups such as patients or staff is done by using rules built by experts. More information can be extracted from these data sets if they are combined and Big Data systems are used to process them.

Real-time processing is used to designate a category where the job outcome is needed as fast as possible, and usually the task itself is not something taking a long time to process. These systems can be categorized as hard or soft. A Hard real-time system is an OS for a nuclear plant or a plane. Tasks must be scheduled and completed fast because otherwise a catastrophe could happen. These systems are usually governed by hard deadlines and the scheduler must insure they are achieved. Soft real-time systems are the ones like hotel booking or video streaming sites [31]. The answers must be delivered fast to the customers, but a delayed frame now and then cannot lead to disastrous results. One article which explores this type of hard real-time scheduling is [32]. In the paper, the authors try to improve the scheduling capabilities of a system by also adding security checks to the incoming jobs. The added module can detect threats brought by snooping, alteration of spoofing and can be easily added to any real-time scheduler. Their security module name SAREC (security-aware real-time heuristic strategy for clusters) integrates with the popular Earliest Deadline First algorithm to create a security aware scheduler named SAEDF. Although the matter of securing the interactions between the users and the cluster infrastructure is important, in our case a large portion of these measures could be implemented in an intermediate cluster proxy module if needed, with little overhead to the job itself. By using a proxy to mediate all user-cluster interactions we can alleviate a large number of security risks. If a user has a malicious intent and manages to submit a job that poses a security risk, runing all jobs in virtual machines on the cluster infrastructure will limit the damages to only the users task.

Another example of real-time processing and scheduling [33]. The authors talk about the problem of soft real-time scheduling in rendering 3D images inside the Google Earth software. The Google Earth software allows one to navigate anywhere in the world and has multiple viewing modes from virtual 3D renderings to satellite imagery. A frame is a static 2D representation, rendered on the screen at a given time. To ensure a smooth navigation experience, at least 60 of these frames must be rendered on the users' screen in a second. When a scheduling deadline is not met, the previous frame is redisplayed causing the application to "stutter". In order to alleviate the problems, the authors have devised a new

[1] "The age of big data" - Steve Lohr, New York Times, 11, 2012.

Please cite this article as: C. Chilipirea et al., An integrated architecture for future studies in data processing for smart cities, Microprocessors and Microsystems (2017), http://dx.doi.org/10.1016/j.micpro.2017.03.004