available at www.sciencedirect.com

ScienceDirect

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

Survey

Streaming techniques and data aggregation in networks of tiny artefacts

Luca Becchetti[a], *Ioannis Chatzigiannakis*[b], *Yiannis Giannakopoulos*[c,*]

[a] *SAPIENZA University of Rome, Dipartimento di Informatica e Sistemistica, Via Ariosto 25, 00185 Rome, Italy*
[b] *Research Academic Computer Technology Institute (RACTI), 1 N. Kazantzaki Str., University of Patras Campus, Rion 26504, Greece*
[c] *University of Athens, Department of Informatics, Panepistimioupolis, 15784 Athens, Greece*

ARTICLE INFO

Article history:
Received 11 June 2010
Received in revised form
28 August 2010
Accepted 15 September 2010

Keywords:
Data streams
Aggregation
Sensor networks
Database management

ABSTRACT

In emerging pervasive scenarios, data is collected by sensing devices in streams that occur at several distributed points of observation. The size of the data typically far exceeds the storage and computational capabilities of the tiny devices that have to collect and process them. A general and challenging task is to allow (some of) the nodes of a pervasive network to collectively perform monitoring of a neighbourhood of interest by issuing continuous aggregate queries on the streams observed in its vicinity. This class of algorithms is fully decentralized and diffusive in nature: collecting all the data at a few central nodes of the network is unfeasible in networks of low capability devices or in the presence of massive data sets. Two main problems arise in this scenario: (i) the intrinsic complexity of maintaining statistics over a data stream whose size greatly exceeds the capabilities of the device that performs the computation; (ii) composing the partial outcomes computed at different points of observation into an accurate, global statistic over a neighbourhood of interest, which entails coping with several problems, last but not least the receipt of duplicate information along multiple paths of diffusion.

Streaming techniques have emerged as powerful tools to achieve the general goals described above, in the first place because they assume a computational model in which computational and storage resources are assumed to be far exceeded by the amount of data on which computation occurs. In this contribution, we review the main streaming techniques and provide a classification of the computational problems and the applications they effectively address, with an emphasis on decentralized scenarios, which are of particular interest in pervasive networks.

© 2010 Elsevier Inc. All rights reserved.

* *Corresponding author.* Tel.: +30 2107275139; fax: +30 2107275114.
E-mail addresses: becchett@dis.uniroma1.it (L. Becchetti), ichatz@cti.gr (I. Chatzigiannakis), ygiannak@di.uoa.gr (Y. Giannakopoulos).

1574-0137/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cosrev.2010.09.007

1. Introduction

Technological advancements and socioeconomic forces have transformed the way in which we live, work and communicate with each other. In this new era, perhaps the most critical variable upon which all our development is based is that of efficiently managing the huge amount of information constantly being generated in various diverse forms and locations. The predominant computational model in modern environments is distributed in nature: many remote devices, possibly different in hardware specifications, are continuously observing and generating huge amounts of data that far exceed their storing, processing and energy capabilities and are organized in dynamically evolving, pervasive network infrastructures that also have limited bandwidth and serving capabilities with respect to the amount and the dissemination of the tasks we are asking them to perform.

In this survey we deal with the algorithmic issues underlying such settings, giving special emphasis on the assumption that our fundamental processing units are *tiny artefacts*, small and usually inexpensive devices with very limited storage, computational power, energy independence and, of course, reliability. In particular, we are interested in being able to efficiently extract crucial statistical information regarding our entire network, in the form of *aggregate queries*. We review important results from the areas of data streaming and database management, in Section 2 describing the fundamental algorithmic techniques of traditional, centralized data streaming. Then, using these as building blocks, we cover distributed computational models in Section 3.

In no way do we consider this survey to be exhaustive. The area of data streaming is very wide and constantly evolving and we refer to other treatments [1–6] and excellent tutorials [7,8] for further consideration.

1.1. Motivation

Our main motivating applications in this survey arise in the field of *sensor networks* [9]. The authors in [10–12] report the deployment of such networks in a wide range of scientific, security, industrial and business applications. Examples include climatological and environmental monitoring, traffic monitoring, smart homes, fire detection, seismic measurements, structural integrity, animal control and habitat monitoring. Apart from sensor networks, other motivating applications include IP routing and network traffic monitoring and analysis, managing large databases, secure and real-time financial transactions and of course, *the Web* itself [1,2,13,3].

1.2. Aggregation

In the scenarios outlined above, single individual values are usually not of great relevance. In fact, users are more interested in the quick extraction of succinct and useful synopses about a large portion of the underlying observation set. Consider, for example, the case of a temperature sensor network. We would like to be able to continuously monitor the entire infrastructure and efficiently answer queries such as "What was the average temperature over the entire terrain during the last 12 h?", or "Are there any specific clusters that have reached dangerously high temperatures?".

As already mentioned and further discussed in Section 1.4 below, trying to collect all data monitored by the sensors would be unrealistic in terms of bandwidth, power consumption and communication intensity. So, the canonical approach is to compute statistical *aggregates*, such as max, min, average, quantiles, heavy hitters, etc., that can compactly summarise the distribution of the underlying data. Furthermore, since this information is to be extracted and combined across multiple locations and devices, repeatedly and in a dynamic way, *in-network aggregation* schemes [14] must be developed that efficiently merge and quickly update partial information to include new observations. Also notice that computing aggregates instead of reporting exact observations can leverage the effect of packet losses and generally network failures, which are common phenomena in wireless networks of tiny artefacts. We deal with such issues explicitly in Section 3.3.

1.3. Traditional vs. sensor network streaming

It is evident that there are two levels of computation and aggregation in distributed settings. At a low level, each sensor observes a stream of data and needs to efficiently extract and maintain information about it. This is essentially the problem of traditional, centralized streaming which has been extensively studied during the last two decades [15–17]. Aggregation is considered with respect to the individual values comprising the data stream, into a concise summary. This is the subject matter of Section 2.

At a higher level, all remote sites should coordinate to combine the partial information computed from each device. Here, aggregation is considered with respect to this merging process of creating summaries that describe the entire infrastructure. Obviously, new challenges are imposed in such distributed settings, which we address in Section 3. It should be clear that this in-network aggregation model generalizes traditional streaming, in the way that a single data stream can be seen as values distributed along a linear-chain topology [18, Section 1.3]. Efficient algorithms for distributed computation that do not make stringent assumptions about the infrastructure topology can be readily used for classical streaming problems.

1.4. Physical restrictions and algorithmic challenges

In the setting of massive data stream computation addressed in this survey, data are observed or produced at a far higher rate than can be locally stored, or sometimes even observed. Their delivery to aggregation or elaboration points requires an amount of in-network communication that far exceeds the power capabilities of sensing devices. Furthermore, the computational complexity of exactly evaluating the statistical aggregates of interest is unrealistic [6]. Considering further that we are interested in scenarios where these functions are performed by tiny devices with extremely limited resources, it is natural to ask for algorithmic solutions and data structures that require storage and update times that are sublinear and often (poly)logarithmic with respect to the size of the observed data, further imposing similar constraints on the amount of communication involving each device. We address these issues in more detail to Sections 2.3 and 3.1.

Download English Version:

https://daneshyari.com/en/article/471532

Download Persian Version:

https://daneshyari.com/article/471532

Daneshyari.com