The Journal of Systems and Software 83 (2010) 2627–2641

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Effective processing of continuous group-by aggregate queries in sensor networks

Chun-Hee Lee[a], Chin-Wan Chung[a,*], Seok-Ju Chun[b]

[a] Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Republic of Korea
[b] Department of Computer Education, Seoul National University of Education, Seoul 137-742, Republic of Korea

ARTICLE INFO

Article history:
Received 16 April 2009
Received in revised form 18 June 2010
Accepted 18 August 2010
Available online 29 September 2010

Keywords:
Sensor network
Group-by aggregate query
Haar wavelet
Two-phase collection

ABSTRACT

Aggregate queries are one of the most important queries in sensor networks. Especially, group-by aggregate queries can be used in various sensor network applications such as tracking, monitoring, and event detection. However, most research has focused on aggregate queries without a group-by clause.

In this paper, we propose a framework, called the G-Framework, to effectively process continuous group-by aggregate queries in the environment where sensors are grouped by the geographical location. In the G-Framework, we can perform energy effective data aggregate processing and dissemination using two-dimensional Haar wavelets. Also, to process continuous group-by aggregate queries with a HAVING clause, we divide data collection into two phases. We send only non-filtered data in the first collection phase, and send data requested by the leader node in the second collection phase. Experimental results show that the G-Framework can process continuous group-by aggregate queries effectively in terms of energy consumption.

1. Introduction

Sensor networks consist of small sensors which have computing and communication facilities. With the advancement of sensor technology, sensors are becoming smaller and more powerful. Moreover, as the price of a sensor becomes low, we expect that a large number of sensors will be used in various sensor network applications.

For example, a volcanologist can use a sensor network to monitor a dangerous active volcanic area. Low-priced sensors can be scattered over the dangerous area from an airplane. Such sensors become a sensor network and monitor the volcano without humans' help. However, sensors have very limited resources (e.g., memory, computation, communication and energy). Among various resources, energy is one of the very important resources since the battery replacement is difficult or impossible in such environments. In sensor networks, since individual sensor readings are raw data, there are many applications using aggregate values. In many cases, the aggregate values of many regional areas are preferred to the aggregate value of the whole area since the aggregate value of the whole area does not provide the detailed information. That is, group-by aggregate queries are useful in sensor networks. Therefore, in this paper, we consider continuous group-by aggregate queries. Due to many shortcomings of the current technology, it is difficult to manage a large number of sensors. As one of the effective methods to deal with many sensors, we can use clustering in sensor networks (Heinzelman et al., 2002; Younis and Fahmy, 2004). Since sensor readings have spatial correlations, spatial clustering of sensors has many benefits. Therefore, we deal with group-by aggregate queries in the environment where sensors are grouped (clustered) by the geographical location. A group-by aggregate query may have a HAVING clause which is a predicate for the aggregation of the group. The queries we consider in this paper are shown in Fig. 1. However, we focus on the query in Fig. 2(a) since processing of queries in Fig. 1 can be extended from the processing of the query in Fig. 2(a). Also, the G-Framework can process local predicates in a straightforward method. Each node checks whether sensor readings satisfy local predicates and makes the bitmap. Then, the node sends only the satisfied data and the bitmap. Therefore, we will not mention local predicates in this paper for convenience of explanation.

Many papers proposed the processing of aggregate queries (Madden et al., 2002; Fan et al., 2002; Considine et al., 2004; Nath et al., 2004; Shrivastava et al., 2004; Deligiannakis et al., 2004; Sharaf et al., 2003, 2004). However, most of them do not consider group-by aggregate queries. Although some papers deal with processing group-by aggregate queries, they do not focus on processing group-by aggregate queries by the geographical location. In this paper, we focus on processing those queries. They can be used in many sensor networks applications such as tracking, monitoring, and event detection. To process them, we assume the following:

- Sensors are grouped according to the geographical location. See Fig. 2(b). A group consists of a leader node and member nodes. A leader node and member nodes are connected in one hop (the

* Corresponding author. Tel.: +82 42 350 3537; fax: +82 42 350 7737.
E-mail addresses: leechun@islab.kaist.ac.kr (C.-H. Lee), chungcw@kaist.edu (C.-W. Chung), chunsj@snue.ac.kr (S.-J. Chun).

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.08.049

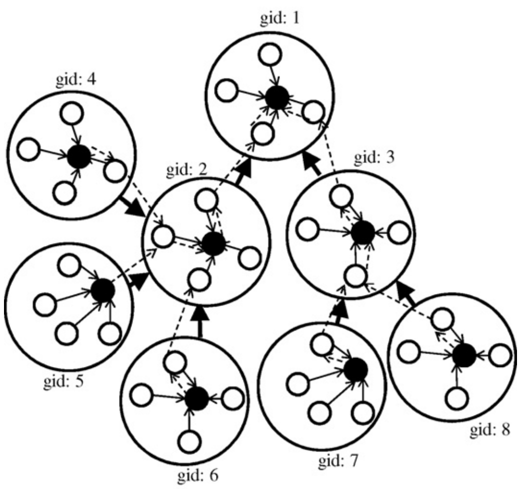
SELECT agg1(attr1), gid
FROM sensor
WHERE local predicates
GROUP BY gid
[HAVING agg2(attr2) op τ]

agg1, agg2: count, min, max, sum, average
attr1, attr2: any attribute in a sensor
op: $>, <, \geq, \leq, =, \neq$

Fig. 1. Query template.

SELECT sum(temperature), gid
FROM sensor
GROUP BY gid
HAVING sum(temperature) > τ

(a) Query



(b) Topology

Fig. 2. Query and topology.

solid lines in Fig. 2(b)). Although we assume the one hop connection, a larger group with multi-hop connections can be handled as discussed in Section 7.
- There is a tree routing topology among leader nodes (the dotted lines in Fig. 2(b)). The thick arrow between groups means the parent-child relationship. We can construct the tree routing topology for the leader nodes by flooding.
- Sensors are synchronized. To synchronize sensors in the G-Framework, we can use the synchronization approach in Ping (2003). In the approach, a master node is chosen as the time coordinator, and broadcasts the time synchronization message. A receiver node takes the message, measures the delay between the master node and itself and synchronizes the time. The approach in Ping (2003) is a lightweight approach (i.e., energy-efficient) and can be applied to multi-hop networks. After a certain time, synchronized sensors are unsynchronized due to various factors. Therefore, we should synchronize sensors periodically.

Fig. 2 shows the query and topology we consider. The query is to monitor the regions when their aggregate values are more than τ. A group is formed according to the geographical location and has a leader node and member nodes. The leader node collects data to compute the aggregation of the group. This query can be used in many sensor network applications. For example, consider building monitoring systems which automatically control the status of the building such as the room air temperature. If the room air temperature is more than a given threshold, we want to turn on the air conditioner. To monitor the status of each room, we install many sensors in the building. We can then group sensors according to the room. All sensors in the same room belong to the same group. The leader node of a room collects sensor readings from member nodes and sends the aggregate value to the base station. In building control systems, the base station receives the aggregate value of each room and controls the air conditioner using the aggregate value.

To effectively[1] process continuous group-by aggregate queries in sensor networks, we consider the following two factors.

- **Approximate processing**: A sensor gets sensor readings from the device. However, no matter how much the device is advanced, there are gaps between real values and sensor readings. Therefore, sensor readings have inevitable errors and small errors are allowed in such environments. Given an error threshold from a user, we will compute aggregate values of groups within the error threshold.
- **Delayed processing**: Continuous queries get results successively. In monitoring applications, a user does not need results immediately. Therefore, delayed results are allowed in such environments. We will compute aggregate values of groups with a delay.

Considering the above two factors, we propose a new framework, called the *G-Framework*, to process continuous group-by aggregate queries. In the G-Framework, we focus on reducing the communication cost since it is the primary factor of energy consumption. We use Haar wavelets to reduce the intra-group communication cost and inter-group communication cost in the G-Framework. Since Haar wavelets reduce data very effectively and are simple, they can be adapted well in sensor networks. To compute the aggregation of a group, sensor readings of member nodes are collected in the leader node. To reduce the intra-group communication cost (i.e., communication cost between the member node and the leader node), in the G-Framework, a member node collects sensor readings during a fixed period instead of sending a sensor reading immediately. Then, the member node compresses them using one-dimensional Haar wavelets and sends important wavelet coefficients to the leader node. The leader node receives wavelet coefficients from member nodes and computes the aggregation of the group.

The aggregate value of each group should be transmitted to the base station effectively. To do that, we use two-dimensional wavelets. The parent group receives the aggregation vectors from the child groups and compresses them using two-dimensional Haar wavelets. By using two-dimensional Haar wavelets, we can send the aggregation vectors to the base station effectively.

A group-by aggregate query may have a HAVING clause. In the G-Framework, we use two-phase collection to process a HAVING clause effectively. To perform two-phase collection, we set the filter condition $v_i \leq F_i$ (v_i is the sensor reading of node i and F_i is the filter value of node i) to each node i of a group according to the HAVING clause. In the first collection phase, a member node sends only sensor readings which are not valid for the filter condition. After receiving data in the first collection, the leader node determines sensor readings to send in the second collection phase.

1.1. Contributions

Our contributions are as follows:

[1] *Effectively* in this paper is used as the meaning of *effectively in terms of energy saving*.