

A New Virtual Select Database Operation for Wireless Sensor Networks

Seungjae Lee*, Changhwa Kim, Sangkyung Kim

Department of Computer Science and Engineering
Kangnung National University
Jibyun-dong, Gangnung-si, Gangwon-do
KOREA
{silveree*, kch, skkim98}@kangnung.ac.kr

Abstract. Sensor networks gather tremendous data from some wide range of area and databases are good to process and manage such tremendous data. However, there might happen some problems on applying database concepts into wireless sensor networks. The reason is because from the viewpoint of wireless sensor networks, situations under which some operations should be performed on non-existing data may occur frequently. For instance, consider the following examples: How can we write a query to get the locations at which the temperature is 22°C , given a sensor network area in which there are some spots at exactly 22°C but there is no sensor node to announce exactly 22°C ? Can we write a query to get the temperature of a spot with no sensor node? In order to solve the above described problems, we propose the new database operation for sensor networks and show how the above queries' results could be obtained using this new operation. This new database operation can provide more effective data management and standard interfaces to application programs for sensor networks. Furthermore, it is also helpful to save a node's energy by reducing the number of communications for processing a query and enhances the robustness of a sensor network system.

Keywords: database, select operation, communication energy, wireless sensor network

1 Introduction

As implementation areas of wireless sensor networks become wider, more research is being done on relational database approaches to wireless sensor networks in order to process and manage tremendous amount of sensing data which is being gathered. However, there might be some problems in directly applying traditional relational database concepts into a wireless sensor network [1,4,5].

Assuming a temperature monitoring wireless sensor network in which temperature sensor nodes are scattered randomly, each node can be described as a point in the whole network area and nodes cannot cover the whole network area continuously. In this situation, if one poses a query to determine the temperature of a point where no sensor node exists, the result would be an empty set because there is no tuple related to the location. And if he or she poses a query to determine locations with 22°C , the

result would be an empty set or few tuples because only sensor nodes which sense exactly 22°C would respond.

The virtual select operation, proposed in this paper, is an operation which can generate non-existing tuples, called to virtual tuples, in the relation. This operation extracts the tuples from the relation if there are tuples that are satisfied with the predicate of the operation in the relation. But if there is no such tuple that satisfies the predicate, the operation generates virtual tuples with real existing tuples, called to the source tuples, of the relation. Generated virtual tuples are tuples created with estimated attribute values by the virtual select operation that could not be generated by combining tuples in the relation or by any other traditional database operations.

Applying a relational database with the virtual select operation, we can pose a query more freely. It enables application code simpler and can provide standard application interfaces. Moreover, it also enables to save node's transmission energy which is a very important research topic on wireless sensor networks. In underwater acoustic sensor networks, energy constraint is more important because acoustic waves need more transmission energy than radio waves and batteries are hard to be recharged or replaced.

The rest of this paper is structured as follows. In Section 2, we present a brief overview of the related works and Section 3 describes the virtual select operator in detail. Section 4 describes how to apply this new operation to wireless sensor networks and the performance analysis is shown in Section 5. Finally, Section 6 concludes and finalizes this paper.

2 Related Works

In a wireless sensor network, database concepts, especially relational database, are used in order to manage tremendous amounts of data effectively. A wireless sensor network can be regarded as a database called to a wireless sensor network database. A tuple, in a wireless sensor network database, is configured with a sensing value, a sensing time, a node's location, etc. A set of sensing values gathered from a class of sensor nodes is treated as an attribute and a set of tuples created by a class of sensor nodes is treated as a relation [1].

On materializing a wireless sensor network database, some work has shown that in-network processing of sensor data is fundamental to achieving energy-efficient communication in wireless sensor networks [1,2,3].

For those who continuously monitor the environment, approximate answers have another feature, called streamed results, that is important for wireless sensor networks. Users can get partial query results and dynamically refine their queries with this feature. This capability, called online aggregation, has been proposed in database literature for large on-line decision support systems. In the wireless sensor network context, users could approach more specific queries [6,7,8].

3 Virtual Select Operation

The virtual select operation generates virtual tuples with existing tuples in a relation. We use σ^v to denote the virtual select operation which is similar with the symbol of select operation in relational algebra. All basis attributes must be appeared in operator's predicate.

The basis attribute values of virtual tuples to be created are defined at predicate of the virtual select operation and can be constant values or ranges. If B_1 is a basis attribute and c_1 and c_1' are constants, constant predicate is declared to be $B_1 = c_1$. In the case of range predicate, five types are available like $c_1 \leq B_1 \leq c_1'$, $B_1 \leq c_1$, $B_1 \geq c_1$, $B_1 < c_1$ and $B_1 > c_1$. $B_1 : c_1$ means $(-\infty, \infty)$. c_1 is called the origin value which is the basic attribute value to create virtual tuples and is explained below in detail.

We call these six predicate types standard predicate types. A virtual select operation with range predicate can be converted into union of virtual select operations with constant predicate.

Predicate in a virtual select operation must have terms on all basis attributes. Predicate which is expressed as logical AND(\wedge) for standard predicate is called to normalized predicate and a virtual select operation expressed as non-normalized predicate can be converted into union of virtual select operation expressed as normalized predicate.

The virtual select operation is expressed as (1) with normalized predicate

$$\sigma_{P_1 \wedge P_2 \dots \wedge P_\beta}^v(\text{relation}) \quad (1)$$

where

$$P_i \left\{ \begin{array}{l} B_i = c_i \\ c_i \leq B_i \leq c_i' \\ B_i \leq c_i \\ B_i \geq c_i \\ B_i : c_i \\ B_i \neq c_i \end{array} \right. \quad c_i, c_i' : \text{constants}, 1 \leq i \leq \beta$$

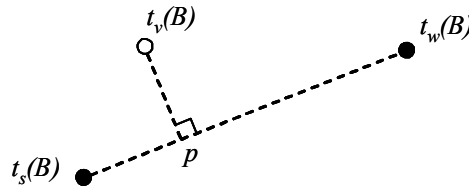


Figure 1. Tuple vectors in Euclidean b-space

Only tuples in which the distance between the tuple vector of the virtual tuple for the basis attribute set B and that of itself is under θ can be source tuples.

A method to calculate the estimate attribute values is like below.

Let $t_s(B)$ and $t_w(B)$ be source tuples for a virtual tuple $t_v(B)$. Then tuple vector vectors of $t_s(B)$, $t_w(B)$ and $t_v(B)$ are described in Euclidean b-space and p is a point where $\overline{t_s(B)t_w(B)}$ and $t_v(B)$ crosses at right angles like figure 1.

Then assuming that the ration of $|\overline{t_s(B)p}|$ and $|\overline{t_s(B)t_w(B)}|$ is the same as that of $|t_s.e_j - t_v.e_j|$ and $|t_s.e_j - t_w.e_j|$ for all $1 \leq j \leq \varepsilon$ where ε is the number of estimate attributes, we can calculate estimate attribute values $t_v.e_j$ using (2).

$$|\overline{t_s(B)p}| : |\overline{t_s(B)t_w(B)}| = |t_s.e_j - t_v.e_j| : |t_s.e_j - t_w.e_j| \quad (2)$$

As $|\overline{t_s(B)p}| = \frac{|\overrightarrow{t_s(B)t_v(B)} \bullet \overrightarrow{t_s(B)t_w(B)}}{|\overline{t_s(B)t_w(B)}|}$, we can write $t_v.e_j$ as (3).

$$t_v.e_j = t_s.e_j - \frac{(t_s.e_j - t_w.e_j) \{ (t_v(B) - t_s(B)) \bullet (t_w(B) - t_s(B)) \}}{|t_s(B) - t_w(B)|^2} \quad (3)$$

Therefore, t_v can be written as (4). If there are more t_s or t_w that satisfy the threshold condition, it iterates the above procedure to create all possible virtual tuples.

$$t_v = (c_1, c_2, \dots, c_b), \quad (4)$$

$$\begin{aligned} t_s.e_1 & - \frac{(t_s.e_1 - t_w.e_1) \{ (t_v(B) - t_s(B)) \bullet (t_w(B) - t_s(B)) \}}{|t_s(B) - t_w(B)|^2}, \\ t_s.e_2 & - \frac{(t_s.e_2 - t_w.e_2) \{ (t_v(B) - t_s(B)) \bullet (t_w(B) - t_s(B)) \}}{|t_s(B) - t_w(B)|^2}, \\ & \vdots \\ t_s.e_\varepsilon & - \frac{(t_s.e_\varepsilon - t_w.e_\varepsilon) \{ (t_v(B) - t_s(B)) \bullet (t_w(B) - t_s(B)) \}}{|t_s(B) - t_w(B)|^2} \end{aligned}$$

4 Implementing Virtual Select Operation on Wireless Sensor Networks

To perform virtual select operation at sensor nodes, it requires two conditions to be satisfied, which are generally valid in wireless sensor networks:

- Each sensor node can communicate directly with nearby nodes.
- The distance between inter-source tuple pairs is less than the transmission range of sensor nodes.

When a sensor node receives a query, it checks whether or not it satisfies predicate itself. If satisfied, it sends a tuple using its own data and finishes the query. Otherwise, it broadcasts the tuple to nearby sensor nodes to create a virtual tuple.

To receive tuples from nearby nodes, it would wait for some brief time which would be stored in the information base of a query layer.

After a sensor node receives tuples from nearby nodes, it checks whether its basis attribute values are the same as those of the received tuple or not. If all basis attribute values are the same, it discards a received tuple. If any of the values is not the same, the node determines whether or not it is the strong sensor node.

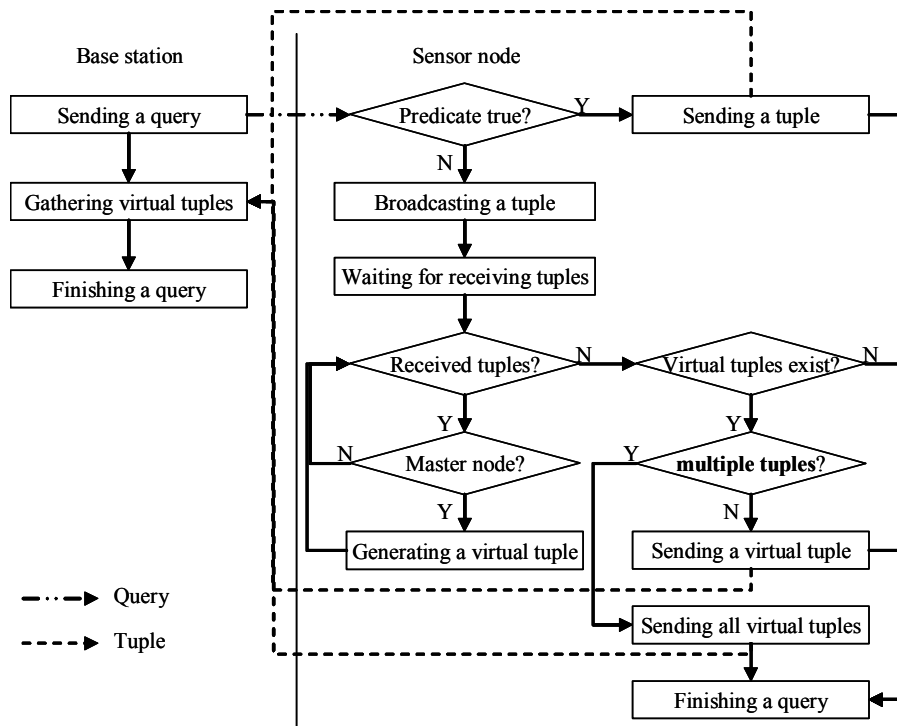


Figure 2. A process diagram of a virtual select operation

A sensor node which provides a strong tuple is called to a strong sensor node and the other sensor node which provides a weak tuple is called the weak sensor node. If two tuples t_1 and t_2 can be a strong tuple at the same time, it is determined by attribute values of the basis attribute set B . Using $t_1(B) \neq t_2(B)$, by definition of a strong and weak tuple, we choose i that satisfies $t_1.b_i \neq t_2.b_i$ at the first time as increasing i from 1 to β and determine the tuple that has the larger b_i as the strong tuple. A strong node might be a master node that generates and sends a virtual tuple.

However, if each node knows its hop count from its own location to the base station, it would be ideal for the sensor node which has the shorter hop count to be a master node. To do this, each node should broadcast its hop count along with its tuple.

If the sensor node is determined not to be a master node, it discards the received tuple. A master node configures a virtual sensor node using data from itself and the other sensor node which sent the tuple. And it also evaluates whether or not the source tuple pair is valid for constraints of θ . If this is not a valid pair, it discards the source tuple pair. If it is a valid pair, a master node generates and sends a virtual tuple to the base station.

Figure 2 illustrates these processes with a diagram.

5 Performance Analysis

A wireless sensor network which consists of n sensor nodes is assumed, such as s_1, s_2, \dots, s_n , l_0 is the amount of overhead for a transmission, and l_t is the length of a tuple in a packet. The hop count from $node$ to a base station is defined as $h(node)$.

If one performs a virtual select operation at a base station, each node should send its tuple to a base station. Therefore, the total amount of transmission data for all sensor nodes is shown in (5)

$$(l_0 + l_t)h(s_i) + (l_0 + l_t)h(s_j) = (l_0 + l_t)(h(s_i) + h(s_j)) \quad (5)$$

where l_0 is the amount of overhead for a transmission and s_i and s_j are the nearest nodes to a location.

But if the query generates virtual tuples in the sensor nodes, two sensor nodes broadcast their tuple to nearby nodes and only a master node transmits a virtual tuple to the base station. Therefore, the total amount of transmission data to send a virtual tuple to a base station is shown in (6) and is less than or equal to (5) where s_i is the master node and $h(s_j) \geq 2$.

$$2(l_0 + l_t) + h(s_i)(l_0 + l_t) = (l_0 + l_t)(h(s_i) + 2) \quad (6)$$

Generally, the hop count from a sensor node to a base station is larger than 2 except at nodes nearby to the base station. Thus it is possible to reduce transmission data by generating virtual tuples in the sensor nodes.

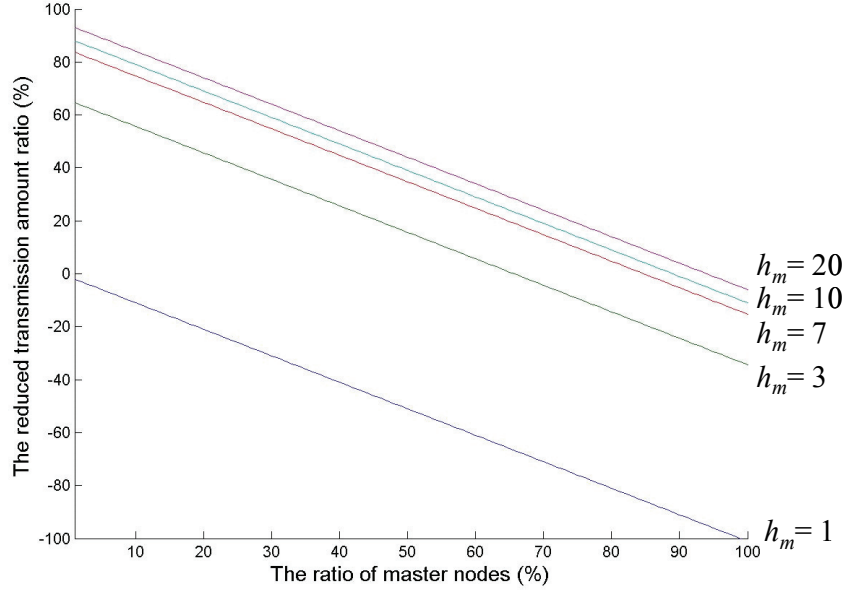


Figure 3. The reduced transmission amount ratio according to the mean value of hops and the ratio of master nodes by the virtual select operation

If the mean value of hop counts h_m is used instead of $h(s_i)$, the equation for the ratio of the reduced amount of transmission data against the original amount of transmission data would be a function of h_m and $\frac{m}{n}$ like (7).

$$f\left(h_m, \frac{m}{n}\right) = 1 - \left(\frac{m}{n} + \frac{1}{h_m}\right) \quad (7)$$

Figure 3 describes graphs for (7) and shows how much transmission data are reduced by the virtual select operation generating virtual tuples in sensor nodes. In this figure, it can be shown that the ratio of the reduced transmission data is larger as the mean value of hops increases and the ratio of master nodes decreases.

6 Conclusion

Sensor nodes are deployed non-continuously in space and its locations are appeared as points. Therefore we can get sensing data at locations where sensor nodes are but cannot at elsewhere.

In this paper, we proposed the virtual select operation to generate virtual tuples. Adapting this operator to a wireless sensor network database, a user can simply pose a

query that needs to estimate some attribute values in the database level. This also can provide a standard interface to application programs and ensure more robust system. In addition, it enables to save energy which is one of the most important resource in wireless sensor networks by reducing the number of data transmission.

In underwater acoustic sensor networks, as acoustic waves need more transmission energy than radio waves and batteries are hard to be recharged or replaced, the proposed operation is more useful on underwater acoustic sensor networks.

A query optimization method for a virtual selection operator and a topology adaptive estimate method are important research topics and those will be explored in the future.

Acknowledgments. This research was supported by the MIC(Ministry of Information and Communication), Korea, under the 2007 ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment)

References

1. R. Govindan, J. Hellerstein, W. Hong, S. Madden, M. Franklin, and S. Shenker. The Sensor Network as a Database. Technical Report 02-771, Computer Science Department, University of Southern California, 2002.
2. John Heidemann, Fabio Silva, Chalermek Intanagonwiwat, Ramech Govindan, Deborah Estrin, and Deepak Ganesan. Building efficient wireless sensor networks with low-level naming. In Proceedings of the Symposium on Operating Systems Principles, October 2001, pp. 146-159.
3. B. Karp and H. Kung. Greedy Perimeter Stateless Routing. In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking(Mobicom 2000), 2000.
4. Philippe Bonnet, Johannes Gehrke, and Graveen Seshadri. Towards sensor database systems. In Mobile Data Management, 2001, pp. 3-14.
5. M. Srivastava, R. Muntz, and M. Potkonjak. Smart Kindergarten: Sensor-Based Wireless Networks for Smart Developmental Problem-Solving Environments. In Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom2001), 2001.
6. J. M. Hellerstein, R. Avnur, A. Chou, C. Hibder, C. Olston, V. Raman, T. Roth, and P. J. Hass. Interactive Database Analysis: The Control Project. IEEE Computer, 32(8), August 1999, pp. 51-59
7. Joseph M. Hellerstein, Peter J. Haas, and Helen J. Wang. Online Aggregation. In Proc. ACM SIGMOID International Conference on Management of Data, 1997.
8. Joshep M. Hellerstein, Ron Avnur, and Vijayshankar Raman. Informix under CONTROL: Online Query Processing. Data Mining and Knowledge Discovery, 4(4), October 2000.