A Distributed Spatial Index for Time-Efficient Aggregation Query Processing in Sensor Networks[1]

Soon-Young Park and Hae-Young Bae

Dept. of Computer Science and Information Engineering, Inha University,
Yonghyun-dong, Nam-ku, Inchon, 402-751, Korea
sunny@dblab.inha.ac.kr, hybae@inha.ac.kr

Abstract. Many applications using sensing data require the fast retrieval of aggregated information in sensor networks. In this paper, distributed spatial index structure in sensor networks for time-efficient aggregation query processing is proposed. The main idea is logically to organize sensors in underlying networks into distributed R-Tree structure, named Sensor Tree. Each node of the Sensor Tree has pre-aggregated results which are the collection of the values of aggregated result for sensing data of the same type of sensors within MBR. If a spatial region query is required, the processing of the query searches the location of the target sensor from the root of the Sensor Tree. And then it finally sends the values of pre-aggregated result of that sensor. By the proposed Sensor Tree aggregation query processing on any region can reduce response time and energy consumption since it avoids flooding query to the leaf sensor or non-relevant sensors in the sensor networks.

1 Introduction

In the past few years, smart sensor devices have matured to the point that it is now feasible as large, distributed sensor networks. Sensors are connected to the physical world which they monitor and collect data. Sensors are connected to other sensors in a through a wireless network, and they use multi-hop routing protocol to communicate with sensors that are spatially distance [9, 12]. Sensors of the same type, for example, temperature sensors, light sensors result the sensing data of the same type which has the same schema. Sensor network consists of lots of sensors and provides opportunities for monitoring information about a spatial region of interest. Sensor data might contain noise, and it is often possible to obtain more accurate result by aggregation data from several sensors. Summaries and aggregates of sensing data are thus more useful than individual sensor readings [4].

Sensors in the sensor networks are usually not connected to a fixed infrastructure, they use batteries as their main power supply, and saving of power is the one of the design issues of a sensor network [10]. But sensor networks can be embedded in a variety of environments. Different applications usually have different requirements from accuracy, power consumption to time-efficient processing. In many spatial applications, for example rescue region control system, spatial queries which gather

[1] This research was supported by University IT Research Project in Korea (ITRC).

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3514, pp. 405 – 410, 2005.

sensing data within a specific region are an essential functionality and require the fast retrieval of aggregated information in sensor networks.

In this paper, distributed spatial index structure in sensor networks is proposed for time-efficient aggregation query processing. The main idea is combining in-network distributed spatial index with the pre-aggregated results, named Sensor Tree. It is to cover the underlying sensors distributed R-Tree using the location of the same type of sensors and to store for each Minimum Bounding Rectangle (MBR), the values of the aggregation function for sensing data of the same type of sensors that are enclosed by the MBR [1, 3]. If a region query is required, the processing of the query is started to search the location of the target sensor from the root of the Sensor Tree and finished by sending the pre-aggregation value of the finding sensor to server which requires the query.

By using pre-aggregation technique of the proposed Sensor Tree region query processing can reduce response time and energy consumption since it avoids flooding query to the leaf sensor or non-relevant sensors in the sensor networks.

The remainder of the paper is structured as follows. Section 2 provides some related work. Section 3 proposes Sensor Tree, which is a distributed index with pre-aggregated results for time-efficient aggregation query processing in sensor networks. Section 4 evaluates the proposed approach by comparison with others. Finally section 5 has concluding remarks of this paper.

2 Related Work

Traditionally, sensors are used as data gathering instruments, which continuously feed a database on server. Each sensor can produce a stream of data about it surroundings. If queries are posed at a powered server, they are flooded to sensors in the networks. Most of all queries over sensor networks are simple and repeatable. In particular, since a message send operation may spend at least 1000 times more battery than a local operation (e.g. sense operation), query processing in sensor network should avoid unnecessary communication as much as possible [6].

There has been much previous work on query processing in sensor networks [7, 8]. Due to the geographical distribution of sensors in a sensor networks, each piece of data generated in the sensor networks has geographic location. And hence to specify the data of the interest over which a query should be answered, each query in a sensor network has a geographical region associated with it [2]. It seems that any kind of index on distributed data requires a hierarchical structure that aggregates information from different region of the networks. Prior work in range query for sensor networks has addressed a number of important issues in constructing such hierarchies. More detailed information can be accessed by top-down traversal of the hierarchy to visit the sensors holding the relevant information [3, 5, 11]. These researches noted the importance of power consumption. They focus on in-network query processing to reduce communication cost and power consumption.

Therefore, there is need for making an efficient access structure on sensor networks in order contact only the relevant sensor nodes for the execution of a query and hence achieve real time response and an accurate result.

3 Sensor Tree: Distributed Index Structure with Pre-aggregated Results in Sensor Networks

In this section, distributed index structure in sensor networks is proposed for time-efficient aggregation query processing. The main idea is to cover the underlying sensors in networks distributed R-Tree with pre-aggregated results, named Sensor Tree. Sensor Tree stores for each Minimum Bounding Rectangle (MBR), the values of the aggregation function for sensing data of the same type of sensors within the MBR. Sensor Tree is built on the sensor of the spatial dimension, therefore its structure is a hierarchical partitioning of the sensor network into rectangle-shaped clusters.

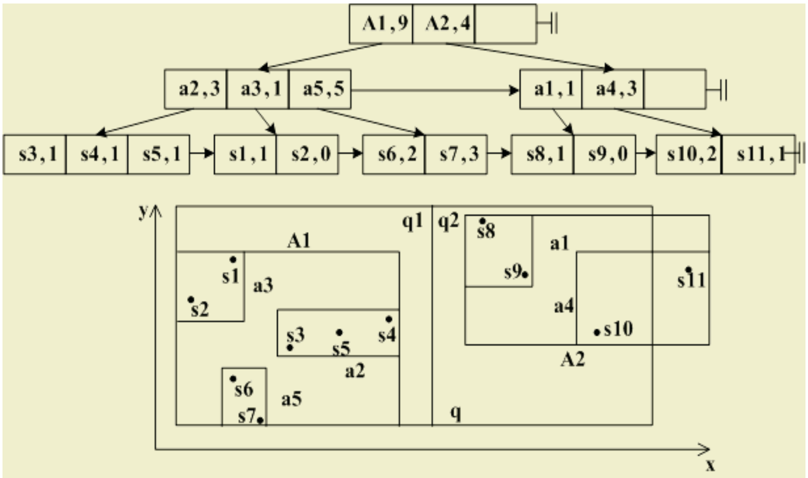

Fig. 1. The Structure of Sensor Tree

It is assumed that every sensor nodes in sensor networks lie in (*x, y*) coordinate space and have been deployed and arranged themselves into a connected topology, protocols for routing of the messages, and the cost of communication between nodes that are distance *d* apart takes O(*d*) amount of energy.

Fig. 1 logically depicts a Sensor Tree which indexes a set of 5 roads *r*1*, r*2*, …, r*5 with MBR is *a*1*, a*2*, …, a*5 respectively when we assume the virtual road is in the rectangle. There are 3 sensors on the road *r*2 and sensing data of sensor *s*3, *s*4*, s*5 are 1, 1, 1 respectively. Therefore the total number of sensing data in *r*3 is 3 and there is an entry (*a*2*,* 3) in the internal node of the Sensor Tree. Moving one level up, MBR *A*1 contains three roads*, r*2*, r*3*, r*5. The total number of sensing data in these roads is 9, and therefore there is an entry (*A*1, 9) at level one of the Sensor Tree.

Each sensor node has a unique identifier, *id*. Each node has a field of communication, which it is capable to send/receive messages. All nodes within this boundary are its immediate neighbors by duplex link. For a node *v*, we denotes *v*'s *r*-neighborhood, *Nbr(v, r)*, as the set of nodes within a radius *r* of *v*. Every nodes *v* maintain a variable *l.v*, level of *v* denoting the highest level of the Sensor Tree that *v* has participated in. Every node in sensor networks cooperates at level 0 of Sensor Tree, only clusterheads of level *i* cooperate for construction of level *i* + 1 of the Sensor Tree. To minimize communication cost, the choice of clusterhead should be tiered into the routing structure. We use a simple clusterhead selection procedure as follows. For every level *i*, *v* maintains a variable *p*, parent node ptr to denote *v*'s clusterhead for level *i*. Dually, *v* maintains *c*, children node ptr, That means, v is the clusterhead of the MBR that contains *c.v(i)*, children of *v* for level *i*. Using this doubly linked structure, any node can query the Sensor Tree. Pseudo code of joining cluster for node *v* at a level *i* is as follows.

```
join_cluster()
{
   if (l.v = i and p.v(i) = nil)
   {
      r <- 1;
      while (Nbr(v, r) < m or p.v(i) != nil)
      {
         if (there exists an i such that
                both u ∈ Nbr(v, r) and l.u = i+ 1)
         {
            p.v(i) <- u;
            c.u(i) <- c.u(i) ∪ {v};
         }
         r <- r + 1;
      }
      if (p.v(i) = nil and (if u ∈ Nbr(v, r) is true
        than so is p.u(i) = nil))
      {
         calculate mbr.v(i) using Nbr(v, r);
         p.v(i) <- v; l.v <- i + 1;
         c.v(i + 1) <- Nbr(v, r);
      }
   }
}
```

A node *v* with *l.v* = *i* executes the join_cluster() if it is not included in the cluster of level *i* + 1, that is *p.v(i)* = nil. In this case *v* first tries to contact a neighboring node *u* with *l.u* = *i* + 1 by searching increasingly larger radii, *r*, and join *u*'s cluster by setting *p.v(i)* = *u*. if such *u* not exists and *v* encounters *n* nodes in level *i* within its *r*-neighborhood, and become the clusterhead of these *n* nodes in one step. If *v* is contacted by another node *u* with *l.u* = *i* + 1, *v* simply joins *u*'s cluster.

During the concurrent executions of join operation by multiple nodes, a clusterhead may end up having more than *N* children. In the splitting cluster, a clusterhead *v* with more than *N* children at level *i* splits its cluster into clusters with number of children greater than or equal to *n* but less than *N*. Split operation is the same in based R-tree. Pseudo code of splitting cluster for node *v* at a level *i* is as follows.

```
split_cluster()
{
   if (c.v(i) > N)
      split(mbr.v(i));
}
```

4 Comparison with Other Approaches

Given a query over a sensor networks, a naive way to run the query will be to simply flood the all of the relevant sensor in sensor network with a query. Each sensor node in the network broadcast the query message exactly once and also remembers the id of the sensor node it receives the query from. If there are n sensors whose sensing regions intersect with the query's region, then using about n message transmissions, a communication routing tree spanning the n sensor could be built within the network. Each node in the built routing tree responds to the query. The responses propagate upwards in the tree towards the root of the query source. This again incurs a cost of n message transmissions, assuming the responses are aggregated at each tree node. Thus, the total communication cost incurred in answering q such queries over the same region is $2qn$ using the naive flooding approach.

Case of in-network query processing to visit only the sensors holding the relevant information, if there are m relevant sensors of the total n sensors, the total communication cost incurred in answering q such queries over the same region is $C + 2qm$, where C, $m \leq C \leq n$, is the communication cost incurred in composing the routing tree by m relevant sensors.

Consider Sensor Tree with pre-aggregated results composed of m sensor. The total cost incurred in executing q queries over the same region will be $D + m$, where D, $m \leq D \leq n$, is the communication cost incurred in composing the Sensor Tree and m is the cost of propagating information from leafs to root of the Sensor Tree in the worst case. If m is substantially less than n, constructing Sensor Tree could result in large saving in communication cost. If there are n sensors, n is 1000, whose sensing regions intersect with the query's region, communication cost incurred in answering query q is as Fig. 2.

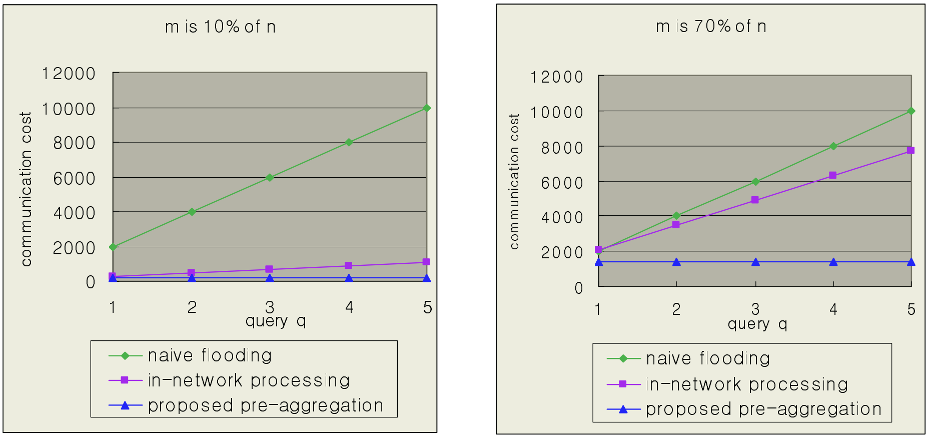

Fig. 2. Comparison of communication cost

5 Conclusion

In this paper, it is focused that efficient execution of region query for aggregation distributed information in sensor networks. It is established that a distributed spatial index structure on sensors of the same type in sensor networks, which is called Sensor Tree. Each node of the Sensor Tree has pre-aggregated result which is the collection of the values of aggregated results for sensing data of the same type of sensors within MBR. Therefore, an aggregation query does not need to access all sensors in networks, since part of the answer is found in the intermediate sensor nodes of the tree.

By using pre-aggregation of the proposed Sensor Tree the query can be answered much more efficiently without processing of aggregation and reduce response time and energy consumption and achieve an accurate response since it avoids flooding query to the leaf sensor or non-relevant sensors in the sensor networks. Our method allows efficient range query in sensor networks.

References

1. A. Guttaman, R-trees: A Dynamic Index Structure for Spatial Searching, Proceedings of SIGMOD 1984.
2. H. Cupta, S. R. Das, and Q. Gu, Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution, Proceedings of MobiHoc 2003.
3. I. Lazaridis and S. Mehrotra, Progressive Approximate Aggregate Queries with a Multi-Resolution Tree Structure, Proceedings of SIGMOD 2001.
4. J. Considine, F. Li, G. Kollios, and J. Byers, Approximate Aggregation Techniques for Sensor Databases, Proceedings of ICDE 2004.
5. J. Gao, L. J. Guibas, J. Hershberger, and L. Zhang, Fractionally Cascaded Information in a Sensor Network, Proceedings of IPSN 2004.
6. N. Demirbas and H. Ferhatosmanoglu, Peer-to-Peer Spatial Queries in Sensor Networks, Proceedings of P2P 2003.
7. S. Madden and M. J. Franklin, Fjording the Stream: An Architecture for Queries over Streaming Sensor Data, Proceedings of ICDE 2002.
8. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks, Proceedings of OSDI 2002.
9. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, The Design of an Acquisitional Query Processor For Sensor Networks, Proceedings of SIDMOD 2003.
10. S. Madden, R. Szewzyk, M. J. Franklin, and D. Culler, Supporting Aggregate Queries Over Ad-Hoc Wireless Sensor Networks, Proceedings of WMCSA 2002.
11. X. Li, Y. J. Kim, R. Govindan, and W. Hong, Multi-dimensional Range Queries in Sensor Networks, Proceedings of SenSys 2003.
12. Y. Yao and J. Gehrke, Query Processing for Sensor Networks, Proceedings of CIDR 2003.