Spatial Query Monitoring in Wireless Broadcast Environment

Koenni Naresh, J. Thangakumar and Durgamahesh Pannem

Abstract Wireless data broadcast is a promising technique for information dissemination that leverages the computational capabilities of the mobile devices in order to enhance the scalability of the system. Under this environment, the data are continuously broadcast by the server, interleaved with some indexing information for query processing. Clients may then tune in the broadcast channel and process their queries locally without contacting the server. It performs location updates only when they would likely alter the query results through monitoring process. Previous work on spatial query processing for wireless broadcast systems has only considered snapshot queries over static data. Here, we use the simple of K-Means clustering algorithm for making clusters of sensor node data in a wireless sensor network. It is used to monitor the objects continuously.

Keywords kNN · WSN · LDIS · LAN · BGI · GPS

Introduction

The use of repetitive broadcast as a way of augmenting the memory hierarchy of clients in an asymmetric communication environment. We describe a new technique called "Broadcast Disks" for structuring the broadcast in a way that provides improved performance for non-uniformly accessed data. The Broadcast Disk superimposes multiple disks spinning at different speeds on a single broadcast

K. Naresh (✉) · J. Thangakumar · D. Pannem
Department of Computer Science and Engineering, Hindustan University, Padur, India
e-mail: nareshkoenni@gmail.com

J. Thangakumar
e-mail: thang.kumar@gmail.com

D. Pannem
e-mail: durgamagesh@gmail.com

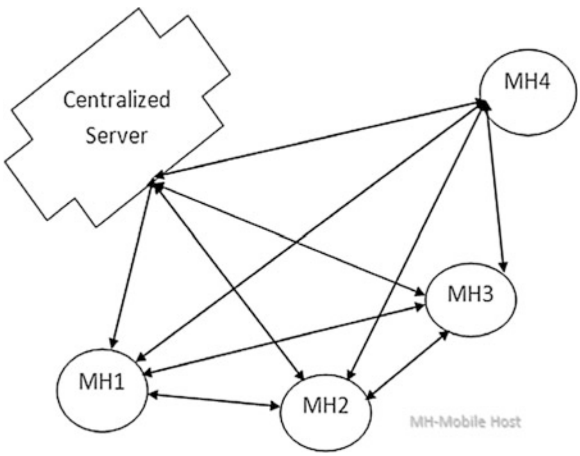
S. Sathiakumar et al. (eds.), *Proceedings of International Conference on Internet Computing and Information Communications*, Advances in Intelligent Systems and Computing 216,
DOI: 10.1007/978-81-322-1299-7_4,

channel—in effect creating an arbitrarily fine-grained memory hierarchy. In addition to proposing and defining the mechanism, a main result of this work is that exploiting the potential of the broadcast structure requires a reevaluation of basic cache management policies. We examine several "pure" cache management policies and develop and measure implementable approximations to these policies. Previous work on location-dependent spatial query processing for wireless broadcast systems has only considered snapshot queries over static data. Suppose moving client wants to know the route from one place to another place. In old GPS application systems client can send the query to server like where am I? the request will go to server and do some process and send the reply to client. Mean while suppose client may reach to another place and he can get the wrong data from the server. It means there is no proper communication between servers and clients. The processing load at the server side increases with the number of queries. In applications involving numerous clients, the server may be overwhelmed by their queries or take prohibitively long time to answer them. Spatial monitoring techniques do not apply to the broadcast environment because they assume that the server is aware of the client locations and processes their queries centrally (Fig. 1).

Proposed System can handle multiple queries at a time. Server will continuously receives the spatial queries from the client side. Because there are more than one client can access the same server at time. In this situation server may react slowly because it has to process lot of queries and filtering. At the client Side, every time it receives the spatial data and continuously accessing server for new data. For that we may have problem with battery consumption in travelling. To avoid this they proposed Air Indexes. The main motivation behind the air indexes is to minimize the power consumption at the mobile client.

Implementation of Air Indexes includes when the query is issued by the client, it tunes to the broadcast channel and it goes to Sleep mode until the next index segment arrives. The client traverses the index and determines when the data objects satisfying its query will be broadcast, The client goes to sleep and returns

Fig. 1 Broadcasting process



to the receive mode only to retrieve the corresponding data objects. We propose an air indexing framework that outperforms the existing techniques in terms of energy consumption while achieving low access latency and constitutes the first method supporting efficient processing of continuous spatial queries over moving objects.

Literature Survey

Area of sensor network is very wide and used in various applications [1]. There has been much work done in data mining, sensor network, data stream etc. but very less work has been done at the combination of these areas. So we introduce the combination of wireless sensor network and data mining to get some interesting and fresh results [2]. Research in WSNs area has focused on two separate aspects of such networks, namely networking issues, such as capacity, delay, and routing strategies; and application issues [3]. The beauty of sensor networking protocols is that they attracted a tremendous amount of research effort For large sensor network the management of sensor database is itself a big task. Query processing techniques have been proposed for acquiring and managing sensor data. One major research goal of this problem in the database community is to efficiently detect outliers in a large-scale database [4]. To collect the data we treat the Sensor networks as databases and SQL queries. Clustering of sensor network is related with network topology, not on the sensor data. In our study we are basically concentrating towards Spatial-Temporal data [5]. Traditional data mining works on Association Rule, looking for patterns of the form: "Customers who buy bread, also buy milk, with probability x %." There is significant literature.

Mobile Location-Dependent Information Services (LDISs) have drawn a lot of attention from wireless data industries in the past few years. This growth in mobile communications presents new aspects to the resource assignment problem as well as new applications [6]. In these services, information provided to mobile users' reflects their current geographical locations. Location dependent data is a data whose value depends on the location. The answer to a query depends on the geographical location where the query originates. Let's consider an example in which a user drives a car and wants to find the nearest gas stations. The user sends a query, such as, "what are the names and locations of the gas stations near to my current location?", using his mobile device [7]. Once the user gets the answer from the server, he will visit the gas stations in order of the nearest to his location based on price. To handle such a query, the positions of the objects and the user must be found. In this paper, we propose the broadcast-based LDIS scheme under a geometric location model. We first introduce the broadcast based location dependent data delivery scheme (BBS). In this scheme, the server periodically broadcasts reports, which contains the IDs of the data items (e.g., building names) and the values of the location coordinates to the clients. The broadcasted data objects are sorted sequentially, based on their location before being broadcasted. Then, we introduce the prefacing scheme in LDIS for the mobile computing environment.

By using the proposed schemes, the client's access and tuning times are significantly reduced. The main contributions of our work can be summarized as follows:

- It is not necessary for the client to wait for and tune into a particular index segment, if it has already identified the nearest object before the index segment has arrived. This technique significantly reduces the access time in the broadcast based LDIS environment.
- The client simply adjusts the value of k when it performs the k-NN query processing.
- The client can also perform the a k-NN query processing without an index segment. In this case, the best access time is obtained, since no index is broadcast along with the file.

System Architecture

Client–server computing or networking is a distributed application architecture that partitions tasks or workloads between service providers (servers) and service requesters, called clients. Often clients and servers operate over a computer network on separate hardware. A server machine is a high-performance host that is running one or more server programs which share its resources with clients. A client also shares any of its resources; Clients therefore initiate communication sessions with servers which await (listen to) incoming requests (Fig. 2).

Wireless Broadcasting

The transmission schedule in a wireless broadcast system consists of a series of broadcast cycles. Within each cycle the data are organized into a number of index and data buckets [8]. A bucket (which has a constant size) corresponds to the smallest logical unit of information, similar to the page concept in conventional storage systems. A single bucket may be carried into multiple network packets (i.e., the basic unit of information that is transmitted over the air). However, they are typically assumed to be of the same size (i.e., one bucket equals one packet). Broadcast process involves the centralized server with multiple number of mobile hosts.

Data Clustering

The term Data Clustering means the process of organizing or arranging a set of objects into groups or clusters in such a manner so that objects within a cluster have the most similarity to one another and the most dissimilarity to objects in

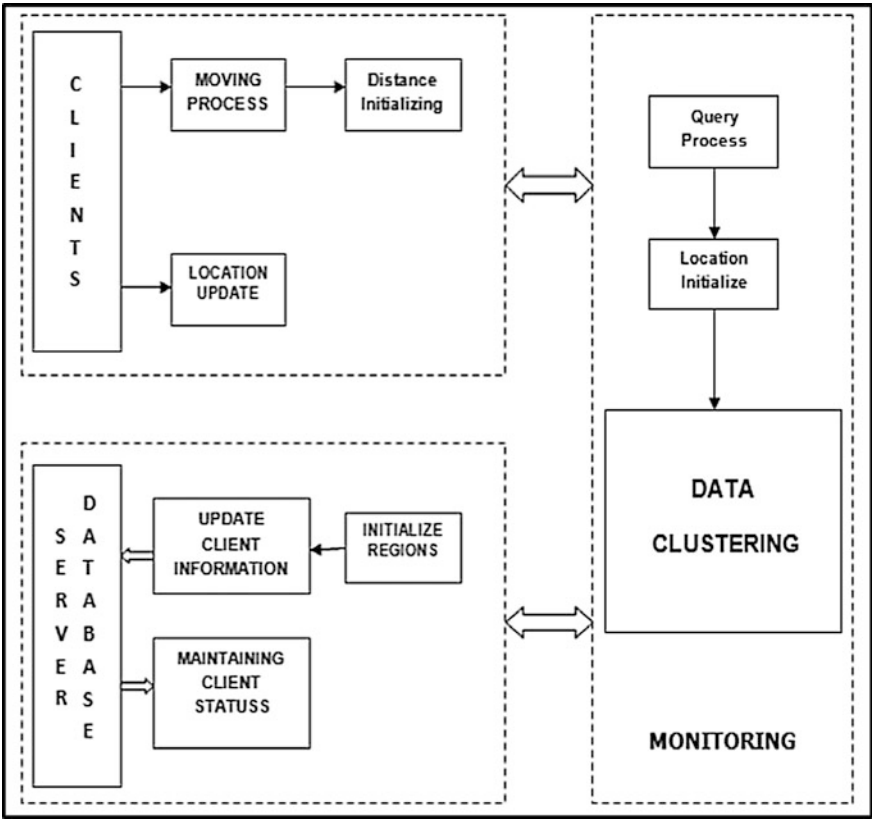


Fig. 2 System architectural diagram

other clusters. Clustering is a useful and basic task in data mining. We generally used it as initial step in data mining for future data analysis. So, here we purpose clustering of sensor data along with their sensor attributes. Here, we use the simple of K-Means clustering algorithm for making clusters of sensor node data in a wireless sensor network.

[9] Data Pre-processing involves cleaning the data by putting in missing values and removing uninteresting data. It may also include summarization and aggregation of the data. This step basically involves preparing the data for analysis. So, firstly we can detect the irregularities in the sensor data and apply pre-processing technique [10]. Pre-processing involves cleaning the data by putting in missing values and removing uninteresting data. It may also include summarization and aggregation of the data. This step basically involves preparing the data for analysis.

Air Indexing

The main motivation behind air indexes is to minimize the power consumption at the mobile client. Although in a broadcast environment, the uplink transmissions are avoided, receiving all the downlink packets from the server is not energy efficient. For instance, the Cabletron 802.11 network card (wireless LAN) was found to consume 1,400 mW in the transmit, 1,000 mW in the receive, and 130 mW in the sleep mode. Therefore, it is imperative that the client switches to the sleep mode (i.e., turns off the receiver) whenever the transmitted packets do not contain any useful information.

Spatial Query Processing

Spatial queries have been studied extensively in the past and numerous algorithms exist for processing snapshot queries on static data indexed by a spatial access method. Subsequent methods focused on moving queries (clients) and/or objects. The main idea is to return some additional information (e.g., more NNs, expiry time, and validity region) that determines the lifespan of the result. Thus, a moving client needs to issue another query only after the current result expires. These methods focus on single query processing, make certain assumptions about object movement (e.g., static in, linear in), and do not include mechanisms for maintenance of the query results (i.e., when the result expires, a new query must be issued).

The possibility of broadcasting spatial data together with a data partitioning index. They present several techniques for spatial query processing that adjust to the limited memory of the mobile device. The authors evaluate their methods experimentally for range queries (using the R-tree as the underlying index) and illustrate the feasibility of this architecture.

Continuous kNN Queries

Consider, for instance, a user (mobile client) in an unfamiliar city, who would like to know the 10 closest restaurants. This is an instance of a k nearest neighbor (kNN) query, where the query point is the current location of the client and the set of data objects contains the city restaurants. Alternatively, the user may ask for all restaurants located within a certain distance, i.e., within 200 m. This is an instance of a range query.

Conclusion

We study spatial query processing in wireless broadcast environments. A central server transmits the data along with some indexing information. The clients process their queries locally, by accessing the broadcast channel. In this setting, our target is to reduce the power consumption and access latency at the client side. We propose an on-air indexing method that uses a regular grid to store and transmit the data objects. We design algorithms for snapshot and continuous queries, over static or dynamic data. To the best of our knowledge, this is the first study on air indexing that

(1) Addresses continuous queries and
(2) Considers moving data objects.

We demonstrate the efficiency of our algorithm through an extensive experimental comparison with the current state-of-the-art frameworks for snapshot queries and with the constant re-computation technique for continuous queries. A challenging problem is to devise cost models for continuous monitoring of spatial queries in wireless broadcast environments. Such models could reveal the best technique given the problem settings, help fine-tune several system parameters (e.g., grid size), and potentially lead to better algorithms. Another interesting direction for future work is to study different types of spatial queries, such as reverse nearest neighbors, and to extend our framework to process their snapshot and continuous versions.

Future Work

Another interesting direction for future work is to study different types of spatial queries, such as reverse nearest neighbors, and to extend our framework to process the snapshots and continuous versions. Another important thing is to add more security for query processing.

References

1. Mittal, R., Bhatia, M.P.S.: Wireless sensor networks for monitoring the environmental activities (2010)
2. Hjaltason, G.R., Samet, H.: Distance browsing in spatial databases
3. Park, K.: An efficient data dissemination scheme for nearest neighbour query processing
4. Acharya, S., Alonso, R., Franklin, M.J., Zdonik, S.B.: Broadcast disks: data management for a symmetric communication environments. In: Proceedings of the ACM SIGMOD (1995)
5. Acharya, S., Franklin, M.J., Zdonik, S.B.: Disseminating updates on Broadcast disks. In: Proceedings of the International Conference on Very Large Data Bases (VLDB'96) (1996)

6. Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R*- tree: an efficient and robust access method for points and rectangles. In: Proceedings of the ACM SIGMOD (1990)
7. Cai, Y., Hua, K.A., Cao, G.: Processing range-monitoring queries on heterogeneous mobile objects. In: Proccedings of the IEEE International Conference on Mobile Data Management (MDM ’04) (2004)
8. Chen, B., Jamieson, K., Balakrishnan, H., Morris, R.: Span: an energy–efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In: Proccedings of the annual international conference on mobile computing (2001)
9. Chen, M.-S., Yu, P.S., Wu, K.-L.: Indexed sequential data broadcasting in wireless mobile computing. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS ’97) (1997)
10. Gedik, B., Liu, L.: MobiEyes: distributed processing of continuously moving queries on moving objects in a mobile system. In: Proceedings of the International Conference on Extending Database Technology (EDBT ’04) (2004)