A Caching Scheme In Location-Dependent Query Processing

Sandi Winn Aye

Abstract : - Wireless networks have become an integral component of the modern communication infrastructure. Mobile database applications through wireless equipment e.g., PDAs, laptops, cell phones and etc. are growing rapidly. Respect to the limitations in mobile environments, it is strongly recommended to minimize number of connections and volume of data transmission from the servers. Caching data item at the client side is a solution. Moreover, the provision of location-dependent information for the same user at different locations is a challenging problem. In this paper, the performance issues of location—dependent queries is addressed, especially nearest-neighbour search. The system investigates a common scenario where data objects (e.g., restaurants and gas stations) are stationary while users/ clients that issue queries about the data objects are mobile. The grid-partition index is used to answer nearest-neighbour queries directly whereas the area within which the answer is valid can be computed. Moreover, a caching scheme (hybrid and semantic caching) is proposed to record a cache item as well as its valid range.

Index Terms: - Grid-partition index, Hybrid caching, KD-tree, Location-dependent query, Mobile database system, Nearest-neighbor (NN) search, Semantic caching, Voronoi Diagram.

————————————————◆————————————————

1 Introduction

Wireless technology is growing rapidly and its beneficial applications which cause use of PDAs, laptops, cell phones and etc. to access data anywhere and anytime, are very common nowadays. Mobile database is such a technology which confronts with some new problems, limitations and challenges (e.g., bandwidth limitations, missing connectivity, unreliable and asymmetric links). Moreover, in a mobile environment, upstream queries (i.e., from client to server) are more resource-consuming than the downstream queries (i.e., from server to client). So, there is a need to reduce the number of trips made to the server. Caching seems to be profitable in mobile environment. There are several types of queries. In this paper, the efficient processing of location-dependent queries and, in particular, a sub-class of queries called mobile nearest-neighbor (NN) search are focused on. A mobile NN search is issued by a mobile client to retrieve stationary service objects nearest to its user. It is an important function for LBSs, but the implementation is difficult since the clients are mobile and queries must be answered according to the client's current locations. For example, "find the nearest restaurant" would return totally different answers to the same user when the query is issued at different locations. If a user keeps moving after he/she submits a query, the problem becomes more complicated because the user's location is changing continuously and thus the results would change accordingly. How to answer a continuous query and provide an accurate answer is an open question. In this paper, a grid-partition index is proposed to support mobile nearest-neighbor search. In this indexing mechanism, both object-based index and solution-based index are used for searching nearest-neighbor data object. In addition, to enhance access efficiency of the system, a new caching scheme in which hybrid caching is combined with semantic caching, is proposed. This caching scheme stores a data object along with the valid spatial scope of the data object. Accordingly, this paper is continued as follows: theory background is studied in section 2. Related work is also studied in section 3. The proposed system is presented in section 4. Finally, this paper is concluded in section 5.

————————————————————

- *Sandi Winn Aye is currently pursuing Ph.D(IT) degree program in University of Technology (Yatanarpon Cyber City), Myanmar, PH-0943028534.*
- *E-mail: sandiwinnayeasb@gmail.com*

2 Theory Background

2.1 Mobile Database System

For any mobile architecture, things to be considered are: users are not attached to a fixed geographical location, mobile computing devices: low-power and portable, wireless networks and mobile computing constraints. Moreover, there are three parties in mobile database system. They are fixed host, mobile units and base stations. Fixed host: It performs the transaction and data management functions with the help of database servers. Mobile units: Portable computers move around a geographical region that is a collection of mobile cells. Mobile hosts retain network connection through the support of base stations. Role of mobile hosts depend on the capacity. Base stations: It is a two-way radio, installation in a fixed location that passes communications with the mobile units to and from the fixed hosts. It is typically used by low-power devices such as mobile phones, portable phones and wireless routers.

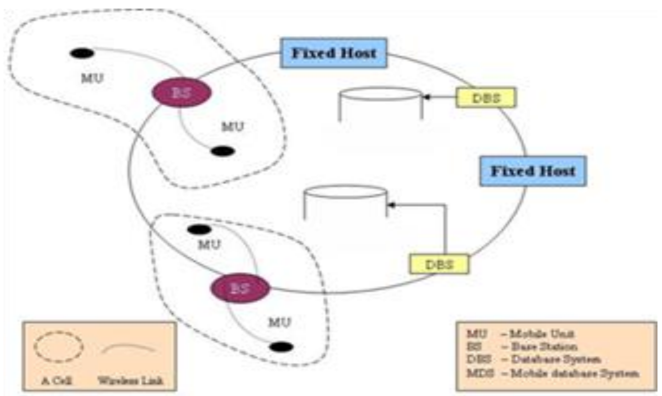


Figure1. Architecture of mobile database system

2.2 Query of Mobile Database System

In mobile environment, there are three types of entities: (i) mobile client that submits a query, (ii) mobile server that processes a query, (iii) mobile object which represents the data targeted by the query. According to these entities there are five query types in mobile environment. They are non-location related query (NLRQ), location-dependent query (LDQ), location aware query (LAQ), continuous query (CQ) and ad-hoc query. Among these query types, in order to support location-dependent queries, some basic requirements must be met. Firstly, the user is located. And then the mobility of moving objects is maintained. Then, the future movement trends are predicted. Moreover, queries are efficiently processed and the boundaries of the precision are guaranteed. According to the mobility of the clients and the data objects to be queried by the clients, location dependent queries can be classified into three categories: (i) mobile client querying static objects (ii) stationary clients querying moving objects (iii) mobile clients querying mobile objects. The application area of location-dependent query can be classified into four categories: (i) location-sensitive information access (ii) object tracking (iii) location model conversation (iv) nearest-neighbor search. In mobile computing environment, the location-dependent information for the same user at different locations is a challenging problem. So, this paper focuses on location-dependent queries. Moreover, nearest-neighbor search is a very important one of problems in location-dependent information services. So, the performance issues of nearest-neighbor search are addressed in this paper.

2.3 Caching

In a mobile environment, upstream queries (i.e., from client to server) are more resource-consuming than the downstream queries (i.e., from server to client), so there is a need to reduce the number of trips made to the server. Caching seems to be profitable in such situation. However, the caching techniques used for traditional database models cannot be applied in this area as it is. Additionally, mobile clients also have limited resources like power, so any of the caching schemes have to be energy efficient and support long and frequent disconnections. Hence, efficiently caching some of the elements that are frequently required by the mobile device can save the scarce resources and give better response time for the client application. There are four caching types: (i) Object Caching: Each mobile client tends to have its own set of hot objects that it accesses most frequently. (ii) Attribute Caching: After the server has evaluated the query submitted by client, it returns only those attributes of the qualified objects that are requested by client. Not caching all attributes saves cache space. (iii) Hybrid Caching: It is a mix of object and attributes caching. So, this caching store attributes of objects that are most frequently used (iv) Semantic Caching: In this caching, the mobile client maintains the semantic descriptions and results of previous queries in its cache. In this paper, hybrid caching is combined with semantic caching to gain performance benefits.

2.4 Grid-partition Index

Traditional nearest-neighbor (NN) search is based on two basic indexing approaches:

1. Object-based indexing: It is constructed based on the locations of data objects.
2. Solution-based indexing: It is built on a pre-computed solution space.

Both approaches exhibit some disadvantages. The object-based indexes incur a low storage overhead, but they rely on backtracking to obtain query results. The solution space typically consists of complex shapes (e.g., polygons), the solution-based index generally has a larger index size than that of the object-based index. So, the grid-partition index which is a hybrid of object-based and solution-based indexes to support location-dependent data. The goal is to combine the advantages of both indexes. The index structure for the grid-partition index consists of two levels. The upper-level index is built upon the grid-cells, and the lower-level index is built upon the objects associated with each grid cell. The upper-level index maps a query point to the corresponding grid cell, while the lower-level index facilitates the access to the objects associated with each grid cell. The advantage is that once the query point is located in a grid cell, its NN is definitely among the objects associated with that grid cell. To construct the pre-computed solution space, Voronoi Diagrams (VDs), which are suitable for nearest-neighbor queries, is used. Thus, preprocessing can be done to construct the corresponding VD of different services. There are three basic approaches to partitioning the search space into grid cells.

1. Fixed partition (FP): It divides the search space into fixed-size grid cells.
2. Semi-adaptive partition (SAP): It adopts the fixed-size partition along one dimension while keeping the partition along other dimension dynamic.
3. Adaptive partition (AP): It adaptively partitions the space using a kd-tree like partition method.

3 Related Works

Xu et al. proposed a new index structure called D-tree [8]. D-tree indexes spatial regions based on the divisions that form the boundaries of the regions. It is an object-based index and has a small index size since it only indexes the necessary information i.e., the position information of objects. However, it requires lots of backtracking in the whole search process. Lee et al. projected a method for answering location-dependent queries in a mobile computing environment [2]. This technique constructs a Voronoi Diagram (VD) on the data objects to serve as an index for data objects. Nevertheless, VD is a solution-space index and its drawback is large index size. Zheng et al. proposed a new index method to support nearest-neighbour queries [6]. Both solution-based index and object-based index have advantages and disadvantages. So, advantages of both indexes are combined in this method. However, this proposed index method is planned for wireless broadcasting. Respect to limited resources in mobile environments, it needs to minimize number of connection and volumes of data transmission from the servers. Caching seems to be very profitable approach in such situation. A semantic cache is proposed to enhance the access efficiency of the service by using Voronoi Diagram [1]. Cache replacement policies for the semantic cache are examined. Several query scheduling policies are proposed to address the inter-cell roaming issues in multi-cell environments. Ali A. Safaei et al. projected a semantic cache schema for continuous k-NN queries in mobile DBSs [3]. Continuous nature of moving in addition to queries raises this fact that caching the previous k-nearest objects in client system's cache will be applicable for the forthcoming queries. The various caching mechanisms are proposed for mobile devices with the emphasis on critiquing the assumptions made in various schemes [4]. These techniques are compared based on their strengths and weakness. The wireless technology has made it possible to achieve continuous connectivity in mobile environment [5]. When the query is specified as continuous, the requesting mobile user can obtain continuously changing result. In order to provide accurate and timely outcome to requesting mobile user, the locations of moving object has to be closely monitored. The problem related to the role of personal and terminal mobility and query processing in the mobile environment are discussed.

4 PROPOSED SYSTEM

This section presents an overview of a caching scheme in LDQ processing. Location-based service (LBS) provides information based on the location information specified in a query. Nearest-neighbor (NN) search is an important class of queries supported in LBSs. Taking the advantages of grid-partition index, hybrid caching and semantic caching, the proposed system can perform to answer location-dependent queries, in which the mobile user issues a query to retrieve stationary service objects nearest to him/user, in mobile computing environment. An index based on grid-partition index is used in the server to support mobile nearest-neighbor search of location-dependent queries. A new caching scheme is proposed to enhance the access efficiency of the service and user mobility. The proposed system consists of two parts: preprocessing in which the region, for each data object, is defined by using grid-partition index in the base server. The second part is caching in which hybrid caching is combined with semantic caching. The previous query answers are cached in the client system's cache. If there is a suitable answer corresponding to the user's query in the client's cache, this query can be answered from that cache without connecting to the server. So that, the number of trips to the server can be reduced and save power. The following figure shows the flow chart of proposed system.

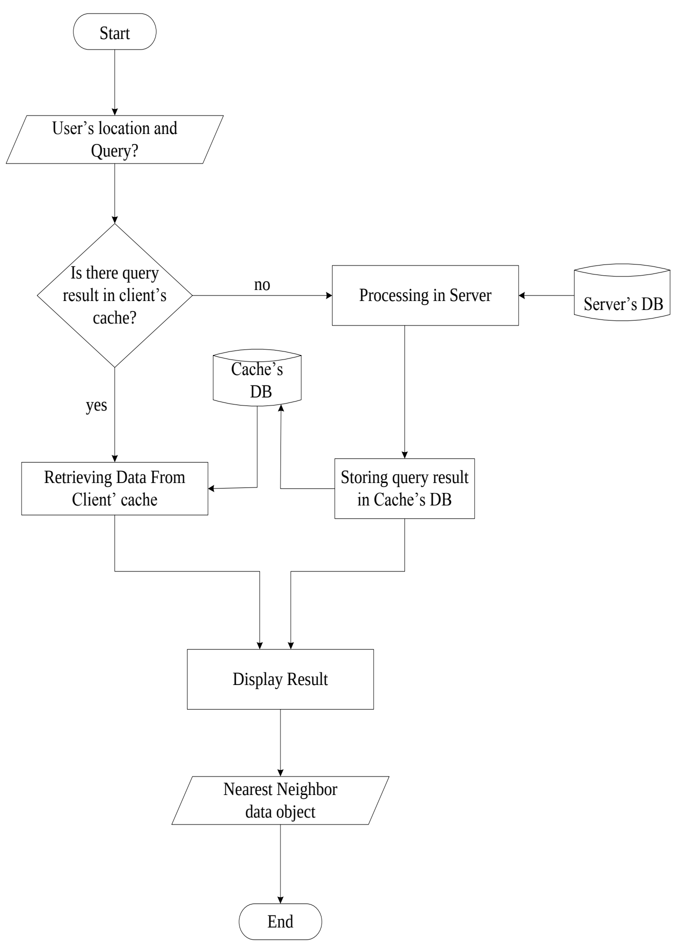
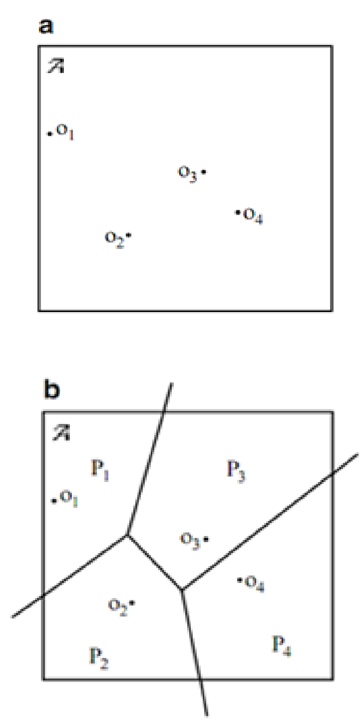


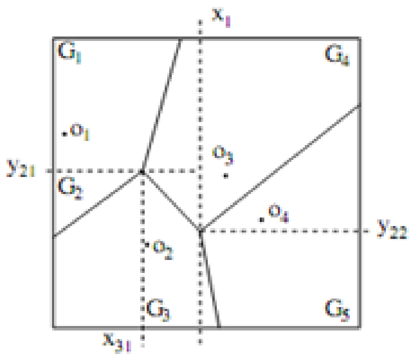
Figure2. Flow chart of proposed system

Firstly, mobile user requests NN query with his/her current location. And then, the client's cache is check to see whether the query answer is available. If there is a suitable cache record corresponding to the location of the user, the most appropriate answer is retrieved from the cache and this answer can be returned to the mobile user quickly. Otherwise, the current location of the user and query are transmitted to the server. The server will first locate this user location in the grid-partition index and then find the nearest-neighbor object to the user. Then, the result is returned to the user. After the user gets the result from the server, a new query answer is inserted in the client's cache.

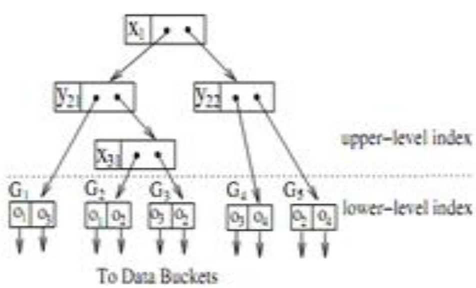
4.1 The Proposed Partition Method

In grid-partition index, both object-based index and solution-based index are used for nearest-neighbor search. Each index has its strength and weakness. Nearest-neighbor query can be answer quickly and effectively by combining strength of these two indexes in grid-partition index. By using grid-partition, the search space can be reduced. Firstly, a Voronoi Diagram (VD), i.e., the solution space of NN queries, on the data objects can be constructed. Secondly, the solution space is divided into a grid cell such that a query point can be efficiently mapped into a grid cell around which the nearest object is located. Thirdly, the grid-partition index stores the objects that are potential NNs of any query within the cell. In grid-partition, adaptive partition is employed. AP adaptively partitions the space using a kd-tree partition method. It recursively partition the search space into two complementary subspaces such that the number of objects associated with each subspace is nearly the same.





(a) AP



(b) Index Structure

4.2 The Proposed Caching

Respect to limitation in mobile environment, it is strongly recommended to minimize number of connections and volume of data transmission from the servers. Caching seems to be profitable in such situations. In our system, a new caching scheme (hybrid caching and semantic caching) for Nearest-neighbor queries in mobile database is proposed. This proposed caching is used to enhance the access efficiency of the service. Caching the previous nearest object in client system's cache will be applicable for the forthcoming queries. Nevertheless, all previous nearest objects can't be cached in client's cache because of limited size of cache. Hot objects and attributes of these objects that it access most frequently are stored in client system's cache. Moreover, these hot data object's descriptions are stored in client's cache. The semantic description of data contains the location of client, the grid cell in which the client locate and data object as <P.x,P.y,G1,O1>. If the position of the client submitting queries is within one of the grid cell, the nearest data object to the client within this grid cell is the answer to the query which can be answered without connecting to the server. Moreover, this description enables to provide partial answers to queries which don't match the cache data exactly. As such, wireless traffic can be reduced and queries may be answer in a disconnected mode.

5 Expected Outcomes

Due to the limited resources in mobile devices and real time requirements of location-based services, it needs to address the difficulties in mobile environment. Caching is to be very suitable solution in such situation. In this paper, a new caching scheme which hybrid caching is combined with semantic caching is used to address the access efficiency of mobile LDQ search. This caching scheme will be a profitable solution for problems occurring in mobile environment. Moreover, a grid-partition method is used to answer queries for finding the nearest services facilities based on mobile user's current location more accurately. Nearest neighbor can be quickly and effectively by combining strength of object-based index and solution-based index in grid-partition method. By using the grid-partition, the search space can be reduced.

6 Conclusion

Mobile database applications through wireless equipment e.g., PDAs, laptops, cell phones and etc. are growing rapidly. Location-dependent queries look for objects in a specific class (e.g., hospital) which is the nearest to the current location of the desired mobile client system. Respect to the limitation in mobile environments, it is strongly recommended to minimize number of connection and volume of data transmission from the servers. Caching is suitable approach. Using caching will significantly reduce the need for connecting and transmitting data from the DB server which is costly. In this paper, a grid-partition index is used to answer queries for finding the nearest services facilities based on mobile client's locations more accurately. Due to the limited resources in mobile devices and real time requirement of location-based services, this proposed caching is used to address the access efficiency of mobile NN search.

Acknowledgment

I specially thank to my Supervisor Dr. Zar Zar Wint. I would also like to thank Rector, Professors and my colleagues from University of Technology, Yatanarpon Cyber City.

References

[1] Baihua Zheng, Wang-Chien Lee and Dik Lun Lee, On Semantic Caching and Query Scheduling for Mobile Nearest-Neighbor Search.

[2] Baihua Zheng and Dik Lun Lee, Semantic Caching in Location-Depenent Query Processing. April 3, 2001.

[3] Ali A. safaei, Mehdi Mallah, Fatemeh Abdi and Shahab Behjati, Semantic Cache Scheme for Continuous k-NN Queries in Mobile DBS, Information Technology Research Journal, 1(1), 2011, 21-32.

[4] Rooma Rathore and Rohini Prinja, An Overview of Mobile Database Caching.

[5] Samidha Dwivedi Sharma, Mobile Database System: Role of Mobility on the Query Processing, International Journal of Computer Science and Information Security, 7(3), 2010.

[6] Baihua Zheng, Jianliang Xu, Wang-Chien Lee and Dik Lun Lee, Energy-Conserving Air Indexes for Nearest Neighbor Search, 2004.

[7] Q.Ren and M.H. Dunham, Semantic caching in mobile computing, 2000.

[8] Jianliang Xu, Baihua Zheng, Wang-Chien Lee and Dik Lun Lee, The D-tree: An index structure for Planar Point Queries in Location-Based Wireless Services. 19th IEEE Int'l Conf. on Data Engineering (ICDE '03), Bangalore, India, 2003.

[9] M. de Berg, M.van Kreveld, M. Overmars and O. Schwarzkopf, Computational geometry: algorithms and applications, chapter 7. (2000)

[10] Baihua Zheng, Grid-partition index: a hybrid method for nearest-neighbor queries in wireless location-based services, VLDB Journal, 2006.