Query Processing with Respect to Location in Wireless Broadcasting

C. Rajeswari and N. Chandrasekaran

***Abstract*—The wireless communication involves a client server communication i.e. the client needs to send a request for performing a process; it can perform only after the response of the server. Large number of request will result the load balance in the server, which cause process delay. It has been resolved by using wireless broadcast client server communication. To communicate with server the client use fee based cellular type network to achieve a responsible operating range. For avoiding these problems, the wireless broadcast model is used. The ad hoc networks are used to share the information among the mobile clients in a peer-to-peer manner for avoiding congestion. The method of single transmission is introduced to support unlimited number of mobile host over a large geographical area. The location based spatial queries often exhibit spatial locality which reduces the time delay. The nearest neighbor candidate can be identified by using the KNN algorithm. After identifying INN candidate, the required data are retrieved using the SBNN algorithm.**

***Index Terms*—Wireless Broadcast, Ad Hoc Network, Peer-to-Peer Network.**

I. Introduction

A wireless network which doesn't depend on any physical connection between two communication entities. Spatial query processing is becoming an integral part of many new mobile applications. LBSQ's refer to spatial queries whose answers relay on the location of the inquires. Wireless communication is used as a term for transmission of information from one place to another without using cable. The mobile phone use two way communication and also known as client server communication. The wireless communication involves a client server communication i.e. the client needs to send a request for performing a process; it can perform only after the response of the server. Large number of request will result the load balance in the server, which cause process delay. It has been resolved by using wireless broadcast client server communication. Broadcasting channels are implemented by which user can act independently and frequently. To communicate with server the client use fee based cellular type network to achieve a responsible operating range. The current location of the user is also not known. For avoiding these problems, the wireless broadcast model is used. In the existing system, congestion occurs due to point to point communication with server for the queries and demand by the client. The ad hoc networks are used to share the information among the mobile clients in a peer-to-peer manner for avoiding that congestion.

Manuscript received April 4, 2012; revised May 18, 2012.
The authors are with the Botho College, Gaborone, Botswana.

II. Peer-to-Peer Cooperative Caching in Mobile Environments

The emergence of robust and reliable peer-to-peer (P2P) technologies now brings to reality what we call "cooperative caching" in which mobile clients can access data items from the cache in their neighboring peers. COoperative CAching Scheme for mobile systems calls COCA [4]. There are two types of mobile clients: low activity and high activity. They are referred to as Low Activity Mobile client/host (LAM) and High Activity Mobile client/host (HAM) respectively. Both LAM and HAM can share their cache. The server replicates appropriate data items to LAMs so that HAMs can take advantages of the LAM replicas. The performance of pure COCA and COCA with the data replication scheme is evaluated through a number of simulated experiments which show that COCA significantly reduces both the server workload in terms of number of requests and the access miss ratio when the MHS reside outside of the service area. The COCA with the data replication scheme can improve the overall system performance in other aspects as well.

III. Group-Based Cooperative Cache

Management for Mobile Clients in a Mobile Environment. Caching is a key technique for improving data retrieval performance of mobile clients. The emergence of robust and reliable peer-to-peer communication technologies now brings to reality what we call "cooperating caching" in which mobile clients not only can retrieve data items from mobile support stations, but also can access them from the cache in their neighboring peers, thereby inducing a new dimension for mobile data caching. This paper extends a Cooperative Caching scheme, called COCA, in a pull-based mobile environment. Built upon the COCA framework, we propose a Group-based Cooperating Caching scheme, called GroCoca, which is build upon COCA frameworks and it is tightly coupled group (TCG) as a set of peers that possess similar movement pattern and exhibit similar data affinity. In GroCoca, a centralized incremental clustering algorithm is used to discover all TCGs dynamically, and the MHs in same TCG manage their cached data items cooperatively. GroCoca is used to reduce the access latency and serve request ratio effectively.

IV. Supporting Cooperative Caching in Ad Hoc Networks

Most researches in ad hoc networks focus on routing, and not much work has been done on data access. A common technique used to improve the performance of data access is caching. Cooperative caching, which allows the sharing and coordination of cached data among multiple nodes, can further explore the potential of eh caching techniques [2]. Due to mobility and resource constraints of ad hoc networks, cooperative caching techniques designed for wired network may not be applicable to ad hoc networks. This cooperative caching techniques is to efficiently support data access in ad hoc networks. It propose two schemes: CacheData which caches the data, and CachePath which caches the data path later, a hybrid approach (HybridCache) which can further improve the performance by taking advantage of CacheDataand CachePath while avoiding their weaknesses.

V. Data Replication for Improving Data Accessibility in Ad Hoc Networks

In ad hoc networks, due to frequent network partition, data accessibility is lower than that in conventional fixed networks. This problem can be solved b replicating data items on mobile hosts. It has three replica allocation methods by assuming that each data item is not updated. In these three methods, the access frequency from mobile hosts to each data item is taken into account and the status of the network connection. It is further extended by considering a periodic updated and integrating user profiles consisting of mobile users' schedules, access behavior, and read/write patterns.

VI. Shared Data Allocation in a Mobile Computing System: Exploring Local and Global Optimization

This data allocation algorithm can utilize the knowledge of user moving patterns for proper allocation of shared data in a mobile computing system. By employing the data allocation algorithms devised the occurrences of costly remote accesses can be minimized and the performance of a mobile computing system is thus improved. The data allocation algorithms for shared data, which are able to achieve local optimization and global optimization, are developed. Local optimization refers to the optimization that the likelihood of local data access by an individual mobile user is maximized whereas global optimization refers to the optimization that the likelihood of local data access by all mobile users is maximized. Specifically, by exploring the features of local optimization and global optimization, algorithm SD-local and algorithm SD-global to achieve local optimization and global optimization, respectively. In general, the mobile users are divided into two types, namely, frequently moving users and infrequently moving users. A measurement, called closeness measure which corresponds to the amount of the intersection between the set of frequently moving user patterns and that of infrequently moving user patterns, is derived to asses' s the quality of solutions provided by SD-local and SD-global.

VII. Information Dissemination via Wireless Broadcast

The advent of sensor, wireless and portable device technologies will soon enable us to embed computing technologies transparently in the environment to provide uninterrupted services for our daily life. An important step towards the realization of this pervasive environment is to be able to disseminate timely and relevant information to the user anytime, anywhere. It consists of a base station, a number of clients and a number of channels [1]. A client can acquire an uplink channel to send a request to the base station and receive the result from a downlink channel. Here it assumes that many users can listen to the same downlink channel to achieve broadcasting from the base station, an uplink channel is dedicated to one client for transmission at any time. The mobile computing environment has a few characteristics such as traditional access latency; mobile clients have limited energy power. In on-demand access, power consumption is dominated by the number of request transmissions since all the client needs to do is to transmit the query and wait for the result, leaving all of the processing work to the server. For broadcast, since the client has to scan the channel for the interested data item, its power consumption can be considered indirectly proportional to the time it is listening actively to the channel, which is referred to as tuning time.

VIII. Infrastructure

Fig 8.1 depicts our operating environment with two main entities: a remote wireless information server and MHs. We are considering mobile clients such as vehicles, which are instrumented with global positioning systems (GPSs) for continuous position information. Furthermore, we assume that the wireless information server broadcasts information in a wireless channel periodically and the channel is open to the public. In addition, there are short-range networks that allow ad hoc connections with neighboring mobile clients. Technologies that enable ad hoc wideband communication include, for example, IEEE 802.11 b/g. Benefiting from the power capacities of vehicles, we assume that each MH has a significant transmission range and virtually unlimited power lifetime. The architecture also supports handheld mobile devices. In Fig. 1 when a MHp issues a spatial query, it tunes in to the broadcast channel and waits for the data. In the meantime, p can collect cached spatial data from peers to harvest existing results in order to complete its own spatial query.

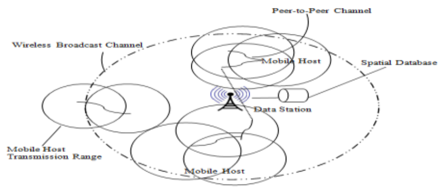

Fig. 1. System environment

Because memory space is scarce in mobile devices, we assume that each MH p caches a set of POIs in an MBR related to its current location. Since the POIs located inside the MBR were obtained from the wireless information server, we define the area bounded by the MBR as a verified region p: V R with regard to p's location.

IX. Spatial Queries

We focus on two common types of spatial queries, namely, KNN queries and WQs. With R-tree-based spatial indices, depth-first search (DFS) and best first search (BFS) have been the prevalent branch-and-bound techniques for processing NN queries.

A. Location-Based Spatial Queries with Data Sharing In Mobile Environments

Location-based queries are of interest in a number of applications, for example, geographical information systems. An example query might be “find the nearest gas station” or “find the three nearest Italian restaurants”. Increasingly such queries are issued from mobile clients. In this study we propose an approach that leverages short-range, ad-hoc networks to share information in a peer-to-peer (P2P) manner among mobile clients to answer location-based nearest neighbor queries. The efficiency of our approach is derived from the observation that the results of spatial queries often exhibit spatial locality. For example, if two mobile hosts are close to each other, the result sets of their kNN queries for a specific object type may overlap significantly. Through mobile cooperative caching, the result sets, query results can be efficiently shared among mobile clients as shown in Fig. 2

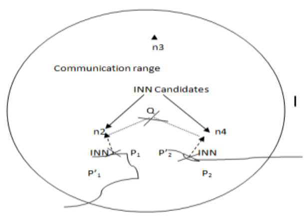

Fig. 2. Example of INN

B. Location-Based Services

Research on data broadcast is typically concerned with scheduling, indexing and caching. In general, broadcast data is assumed to be location independent. Research on data broadcast is typically concerned with scheduling, indexing and caching. In general, broadcast data is assumed to be location independent. In mobile computing where users move around, location becomes an important dimension of data peer-to-peer result sharing. The answer to a query depends not only on the data values but also on the location where the query was issued. These queries are called location-dependent queries (LDQs). The data involved in answering LDQs are called location-dependent data (LDD).

C. Sharing Based Nearest Neighbors Queries

Fig. 3 shows an example of an on-air KNN query based on the Hilbert curve index structure. At first, by scanning the on-air index, the k-nearest object to the query point is found, and a minimal circle centered at q and containing all those k objects is constructed [3]. The MBR of that circle, enclosing at least k objects, serves as the search range. Consequently, q has to receive the data packets that cover the MBR from the broadcast channel for retrieving its k-nearest objects. As shown in Fig. 3, the related packets span a long segment in the index sequence, that is, between 5 and 58, which will require a long retrieval time. The other problem with this search algorithm is that the indexing information has to be replicated in the broadcast cycle to enable twice scanning. The first scan is for deciding the kNN search range, and the second scan is for retrieving k objects based on the search range. Therefore, we propose the Sharing-Based Nearest Neighbor (SBNN) query approach to improve the preceding on-air kNN query algorithm.

D. Broadcast Channel Data Filtering

Under most conditions, there are verified and unverified entries in H when the NNV method cannot totally fulfill a KNN query. For applications that require accurate NN information, we can utilize the partial results to calculate data packet search bounds from the entries in heap H to speed up the on-air NN search process. The heap H is in one of the six different states after an MH has executed the NNV mechanism without retrieving k verified objects State 1. H is full and contains both verified and unverified Entries State 2. H is full and contains only unverified entries. State 3. H is not full and contains both verified and unverified entries. State 4. H is not full and contains only verified entries. State 5. H is not full and contains only unverified Entries State 6. H contains no entries. In state 1, there may exist some POIs that are closer q compared with the last element in H. Hence, we can consider the last entry of H as the final candidate NN in the NN search and utilize its distance as the search upper bound. In addition, the distance attribute dv of the last verified entry can be another bound, that is, the search lower bound. Since we are certain about the POIs within the circle region Ci with radius dv and center point q, q does not have to receive any data packet that contains objects completely covered by Ci. Conversely, when H is full and contains just unverified entries, we can infer only the upper bound (State 2). In States 3 and 4, after the MH performed the NNV algorithm, there have been merely less than k POIs found. Therefore, we can only infer the lower bound from the distance attribute of the last verified element in H.

E. KNN Algorithm

This algorithm used to find the Nearest Neighbor and these detail stored to the Heap data structure table. The KNN algorithm P denotes the Peer nodes responding to the query request issued from q the value of Merged Verified Region (MVR) is arrived as zero. Let IP denote the data collection by q from j inside the $M_{VR.}$ User request is checked whether it is inside the region it also user satisfaction all the information from KNN candidate for shorted by user q.

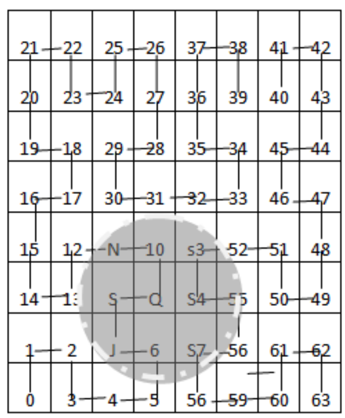

Fig. 3. An on-air kNN query example. The numbers represents index values.

The value of distance between user and surface edge is calculated where e_s is the edge that has the shortest distance to q among all the edged of M_{VR} I value assigned as one. If k elements in H are all verified by NNV, the kNN query is fulfilled. There will be cases when the NNV method cannot fulfill a kNN query. Hence, a set that contains unverified elements is returned. These values maintain into data structure of the Heap H.

POI	Verified?	Distance to q [miles]	Correctness probability	Surpassing distance (r^1-r)
0_1	Yes	2	100%	-
0_5	Yes	3	100%	-
0_4	No	5	55%	2
0_3	no	6	40%	3

The Data Structure of the Heap H

F. SBNN

SBNN is used to retrieve the data into broadcast channel. The value of H is assigned to the kNN algorithm data structure table. If k verified NN's have been retrieved are heap is full and q accepts approximate results. If H is not full or q disallows any approximate result. Utilized search upper and lower bounds to improve the on-air query. The union of H and kNN query results retrieved from the updated on-air query. Then finally the H value is returned.

X. Conclusion

This paper presented a novel approach for reducing the spatial query access latency by leveraging results from nearby peers in wireless broadcast environments. Significantly, our scheme allows a mobile client to locally verify whether candidate objects received from peers are indeed part of its own spatial query results set. The experiment results indicate that our method can reduce the access to the wireless broadcast channel by a significant amount, for example, up to 80 percent, in a dense urban area. This is achieved with minimal caching at the peers. By virtue of its P2P architecture, the method exhibits great scalability; the higher the mobile peer density, the more the queries answered by peers. Therefore, the query access latency can be markedly decreased with the increase in clients.

References

[1] B. Zheng and D. Lun Lee, "Information Dissemination via Wireless Broadcast," *Comm. ACM*, vol. 48, no. 5, pp. 105-110, 2005.

[2] L. Yin and G.Cao, "Supporting Cooperative Caching in Ad Hoc Networks," *IEEE Trans*. Mobile Computing, vol. 5, no. 1, pp. 77-89, Jan. 2006.

[3] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest Neighbor Queries," *Proc. C. M SIGMOD '95*, pp. 71-79, 1995.

[4] C. Y. Chow, H. Va Leong, and A. chann, "Peer-to-Peer cooperative Caching in Mobile Environment," *Proc. 24th IEEE Int'l Conf.* Distributed Computing Systems Workshops (ICDCSW'04), pp.528-233.