Location-Dependent Query Results Retrieval in a Multi-cell Wireless Environment

James Jayaputera and David Taniar

School of Business Systems Monash University,
Clayton Vic 3800 Australia
{James.Jayaputera, David.Taniar}@infotech.monash.edu.au

Abstract. The demand of information services is popular in recent years. However, the requested of correct answer in a mobile environment needs to have more attentions. This is due to the scope of query depends to the user location. In this paper, we propose an extension approach to handle the situation where a mobile user misses a query result at current time and expects to receive a next query result in the next interval time. The aim of this extension approach is to avoid redundant process in order to get a new query result. We show the efficiency of our proposed algorithm by giving some different examples and evaluations.

1 Introduction

Location-Dependent Information Service (LDIS) is one type of applications to generate query results based on the location of users issuing queries (requesters) [1, 2, 3]. It implies whenever users change their locations while they are sending queries, the query results have to be relied on the receiving location of the users receiving queries. Location-Dependent Query (LDQ) is one type of queries based on the data found on that particular location [4, 3]. Hence, the expected results of LDQ must accurate and depend on the new location of user.

In our past papers [5, 6], we proposed an approach to retrieve query results for LDIS applications. The query result retrieval approach allowed that method to retrieve the results produced based on the locations users requesting queries. However, that approach deals if users freely move within one cell, where a cell is an area covered by one base station (BS). [1]

In this paper, we propose an extension algorithm from our previous works. This extension algorithm is to retrieve query results in multi cells. The aim of this paper is to retrieve query results from multi cells accurately. For example, users send queries from current cell and travel with constant velocities and directions. Since a BS covers only a certain area, the users can move from one cell into another. However, delays might occur during this period, which results of

[1] A Base Station is a static host that does an address translation and message forwarding from a static network to wireless devices and vice-versa [7].

J. Cao et al. (Eds.): ISPA 2004, LNCS 3358, pp. 49–53, 2004.

handover or others delays (such as transmission or processing). Handover is a process to transfer an ongoing call from one cell to another as a user changes the coverage area of a cellular system [8].

To simplify our discussion, it is assumed as follow: a geometric location is represented as two-dimensional coordinates, users travel on steady velocities and directions, every BS has knowledge about its neighbours and the expected time to leave current BS, the predicted locations are known before receiving query results and there are no errors in all partials data retrieved. Furthermore, the handover time is ignored since it does not make any changes towards the prediction of users' location. The delay occurred is static instead of variable.

The rest of this paper is organized as follows. In next section, some related works of this paper are presented. In section 3, our proposed algorithm will be discussed and later, examples will be shown. In section 4, we show the performance of our proposed algorithm. Finally, the last section will summarize the contents of this paper.

2 Related Work

In this section, we review the existing studies on those two aspects. Related works to this paper is works have been done in query results retrieval for LDIS applications, including how to retrieve query results from one Base Station and multiple Base Stations while a user is moving from one to another area [5, 6, 9, 10]

Efficiency query result retrieval in one BS for LDIS application has been discussed in our past papers [5, 6]. Figure 1 shows an illustration of our proposed algorithm to retrieve query results within a single BS. Let us considers, two locations: A and B. A user travels to east from A to B at speed 2 is sending a query, "retrieve all vending machines within 1 km from my current location". When the user accepts the query results, the query results must reflect to all vending machines which are located 1 km away from B since the user is not on A anymore. However, the user is only interested with the query results that

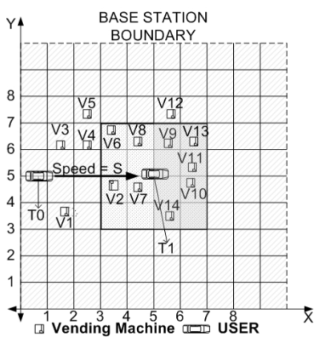

Fig. 1. Query Result Retrieval in a single BS

have not been passed (shaded area). Therefore, the valid query results are the vending machines (V9, V10, V11, V13, V14).

In work done by Sistla et al [10], the location of a moving object is conducted as dynamic attributes which is divided into three sub-attributes: function, updatetime and value. The advantage of their work is a new predication location can be found by using function of time. Therefore, we adopt their method to calculate the prediction location in our approach.

Our work is similar to their works; however, the focus of our work is to retrieve query results from multiple cells. The purpose of our work is to get query results accurately and fast.

3 Query Results Retrieval: Propose Algorithm

In this section, we propose an algorithm for query results retrieval in multicell for location-dependent. In retrieving query results, users can either stay in same location or move to other locations. We are only interested in moving users, however, users travel with variable velocity and directions are not discussed here. The aim of our propose algorithm is to retrieve correct query results from servers where the query scope is crossing the area of BSs.

Our proposed algorithm for query results retrieval from multiple BS is shown in figure 2. *Current_BSID* represents the current BS identification number. BS_{scope} is referred to endpoint coordinates that performed a boundary of BS. In our case,

```
Algorithm: Query_Processing_for_Multi_BS
Input: Query Scope, Number of Neighbor Base Stations, Available Base Stations
Output: result
Begin
  Query_scope ← Scope of query
  BS_scope ← current BS station scope
  BS[1..n] ← online Base Stations
  current_BSID ← ID of current base station
  result ← Get_Result(current_BSID);
  While intersection (Query_scope, BS_scope) is true
     current_BSID ← next neighbour BS ID of the current BS
     BS_scope ← get_scope(current_BSID)
     Query_scope ← Query_scope.MAX - BS_scope.MIN
     /* Append results retrieved to end of records */
     result ← result + Get_Result(current_BSID);
  End loop
  Return result
End Query_Processing_for_Multi_BS
```

Fig. 2. The proposed approach

the number of endpoints used is four since we assume that the scope of BS is a square. All online BSs are stored into a collection, called *BS[1..n]* where n is a number of online BSs. $Query_{scope}$ is four endpoint coordinates that represents a scope of user query.

After the parameters initialization, the current BS checks whether the scope of query is intersect the scope of current BS. It generates query results in the current BS and stores the query results into *result* parameter. If there is no intersection, only the query results in the current BS are returned.

If there is any neighbour BS straight to the current BS, the neighbour BS becomes the current BS. The scope of the current BS is generated and then, the query scope is deducted against the minimum endpoint coordinates of parameter BS_{scope}. The query results within the query scope are generated. These processes keep repeating until there is no any intersection between query scope and BS scope. Then, the query results are forwarded to the user.

4 Performance Evaluation

After we discussed our proposed algorithm, evaluations on our proposed algorithms are given in this section. The objective of our evaluations are to examine situations whether our algorithms can handle situations to retrieve query results from multiple cells efficient and accurately. First, we give examples to simulate our evaluations. Then, we evaluate our examples given. The evaluation results are given at the end of this section show efficiencies of our propose approach.

Figure 3 shows processing time to process one query. The graph does not show straight line graph, because processing time to generate an answer for every query is different from one to another. The processing time to generate query results for query number 45 is the longest since there are a number of users entering cells. It can also be caused by more common data in the server. In contrast, the processing time for query number 10 is the shortest since the data and users entering the cell are rare.

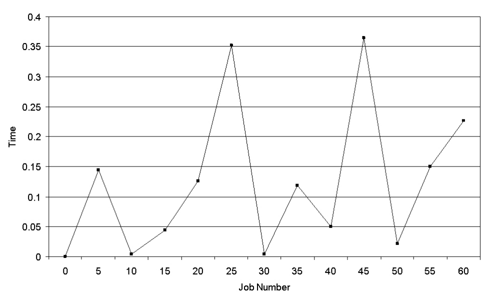

Fig. 3. Processing time for one query

5 Conclusion

In this paper, we have shown how to retrieve query results in multi cell. In early section, we give our motivation on query results retrieval in multi-cell. Afterwards, we propose our algorithm followed by analyses. We assume that the user travels on steady speed and direction. Whenever there is any intersection between query scope and BS scope, it implies the query results are within multiple cells. Therefore, the query scope must be deducted against the minimum boundary of next BS in order to process the remaining query scope inside the next BS. We also deal with delay time if any. However, we do not consider variable value of delay time. When there is any fixed delay time, it is added to the value of retrieval time. Our experiments results show accurate value. In addition, our experiments results show reasonable time to answer the user query.

References

1. Baihua, Z., Lee, D., Xu, J.: Data management in location-dependent information services. IEEE Pervasive Computing **1** (2002) 65–72
2. Tang, X., Xu, J., Lee, D.: Performance analysis of location dependent cache invalidation schemes for mobile environments. IEEE trans. on Knowledge and Data Eng. **15** (2003) 474–488
3. Zheng, B., Xu, J., Lee, D.: Cache invalidation and replacement strategies for location-dependent data in mobile environments. IEEE Trans. on Computers **51** (2002) 1141–1153
4. Dunham, M., Kumar, V.: Using semantic caching to manage location dependent data in mobile computing. Proc. of the sixth annual Int'l Conf. on Mobile Computing and Networking (2000) 210–221
5. Jayaputera, J., Taniar, D.: Defining scope of query for location-dependent information services. Int'l Conf. on Embedded and Ubiquitous Computing (2004) 366–376
6. Jayaputera, J., Taniar, D.: Query processing strategies for location-dependent information services (to appear jan 2005). Int'l Journal of Business Data Comm. and Networking (2004)
7. Goodman, D.J.: Wireless Personal Communications Systems. Addison-Wesley Wireless Communications Series (1998)
8. Markopoulos, A., Pissaris, P., Kyriazakos, S., Sykas, E.: Efficient location-based hard handoff algorithms for cellular systems. NETWORKING 2004, Third Int'l IFIP-TC6 Networking Conference (2004) 476–489
9. Kahol, A., Khurana, S., Gupta, S.K.S., Srimani, P.K.: An efficient cache management scheme for mobile environment. Proc. of the 20th Int'l Conf. on Distributed Comp. Systems (2000) 530–537
10. Sistla, P., Wolfson, O., Huang, Y.: Minimization of communication cost through caching in mobile environments. IEEE trans. on Parallel and Distributed Systems **9** (1998) 378–389