

Mobile Client-Server System for Real-time Continuous Query of Moving Objects¹

Young-Baek Kim*, Young-Choon Kim**, Hae-Jong Joo*** and Sang-Yong Rhee*+

*Dept. of Computer Science and Engineering, Kyungnam University, Changwon, Korea

**Dept. of Car Engineering, Kongju National University, Kongju, Korea

***Dept. of hunic, Dongguk University, Seoul, Korea

Abstract

In this paper, a Mobile Continuous Query Processing System (MCQPS) is designed to solve problems related to database hoarding, maintenance of shared data consistency, and optimization of logging. These problems are caused by weak connectivity and disconnection of wireless networks inherent in mobile database systems under mobile client-server environments. We show the superiority of the proposed MCQPS by comparing its performance to the Client-Intercept-Server (C-I-S) model. In addition, several experimental results show the effectiveness of our proposed indexing structure and methodology for real-time continuous queries.

Key Words : Mobile Client-Server System, MCQPS(Mobile Continuous Query Processing System), Real-time Continuous Query

1. Introduction

Many researches are going on with regard to issues and problems related to mobile database systems, which are caused by the weak connectivity and frequent disconnection of wireless networks, the mobility and the portability of mobile clients[1-3],[7]. Mobile computing satisfies user's demands for convenience and performance to use information at any time and in any place, but it has many problems to be solved in the aspect of data management [1,3].

The purpose of this paper is to propose a new MCQPS in order to solve problems related to database hoarding, maintenance of shared data consistency and optimization of logging, all of which are caused by the weak connectivity and frequent disconnection of wireless networks inherent in mobile database system under mobile client-server environments. A new mobile client-server system including is to be proposed here, which has index structure and processing functions to respond to location information changing of moving objects in real-time and to provide the optimized services in response to their continuous queries, especially when they request continuous queries to get neighboring location information while moving.

This paper has the following configuration. At section 2, for the theoretical approach of this paper, definitions on problems caused by properties of mobile client-server structure and mobile database system will be considered. At section 3, components, design and ways of securing reliability of systems including MCQPS to solve problems of mobile database environments will be described. At section 4, for demonstration

of effectiveness of MCQPS, we will prove that the proposed system is superior to C-I-S(Client-Intercept-Server) by comparing performances of two systems. In addition, we will carry out various experiments to prove the effectiveness of indexing structure and methodology proposed for the real-time continuous queries. At section 5, we will make a final conclusion.

2. Related Research

2.1 Structure of Mobile Client-server Model

The general instruction delivery structure under mobile environment at fig.1 shows the model in which application client sends requests to server on wired network via wireless network, while C-CA-S model introduced by Coda[1] is used on servers with small UNIX file systems and large clients, this client agent(CA) carries out, on behalf of mobile client, operations of files systems performed normally by server.

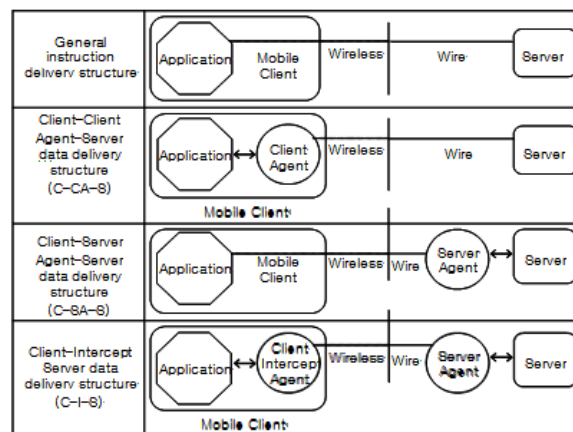


Fig. 1. A Existing Mobile Client-Server Model

Manuscript received Feb. 21, 2011; revised Jun. 1, 2011.

¹This work was supported by Kyungnam University Foundation Grant in 2010.

+ Corresponding author

Server agent(SA) in C-SA-S model is like a proxy carrying out interfaces to server in wired network, which proxy is called server agent. Agent has been used as such on the client or server to improve the functionalities of the existing client-server model, however it is not always easy to modify the application of server or client to communicate with the client agent(CA) while in motion. In order to solve this dilemma, C-I-S model is introduced in which agents on both client and server deal with all communications of wired and wireless network between them.

2.2 Synchronization of the Existing Mobile Database System

Mobile database companies adopt the synchronization scheme in which the whole synchronization is achieved by placing the synchronization server between the central database and each of the mobile database. They adopt the synchronization scheme in which either synchronization between each of the mobile database is not considered, or a new synchronization server is placed between each of the mobile database for synchronization [5]. Most of mobile database companies consider the mobile database as the extension of existing database and use one central sync server to maintain the consistency with center static database[5].

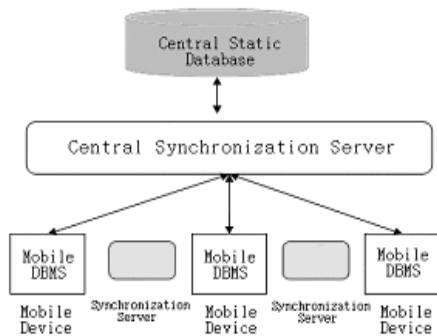


Fig. 2. The Existing Synchronization of Mobile Database System

2.3 Property of mobile database system

Three elements should be considered when designing the database suitable for the mobile database environment as shown in fig.3[1,7]. First, the limits which mobile equipments have fundamentally in wireless network should be overcome. Second, the availability and currency of data in the database application should be guaranteed. Third, data represented in the mobile equipments should maintain consistency with the data used from data server or data warehouse.



Fig. 3. Various Consideration of Mobile Database Environments

2.4 Continuous nearest query for moving object

Moving object means the spatial object which changes its location information as time flows. As the query of such a moving object may produce invalid results according to its location, new results are required every time when location is changed. Thus continuous queries need to be generated as location is changed, and the method of verifying the validity of results is also needed [9,10].

To deal with the continuous nearest queries effectively, we propose the method [9] by which Voronoi cell of static object sets is stored in R-tree after being calculated in advance. Voronoi cell is the region formed by perpendicular bisectors of objects neighboring p_i , the nearest neighbor object of moving object in this region is p_i .

Network burden can be diminished by decreasing the occurrence counts of continuous queries through proving the validity of query results based on Voronoi cell.

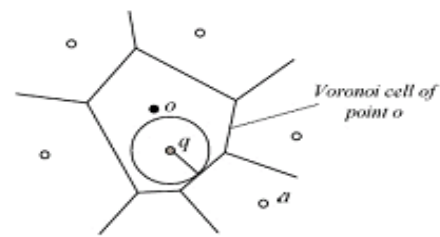


Fig. 4. Voronoi Diagram

3. MCQPS Client-Server system configuration

The new mobile client-server system proposed in this paper consists of C-I-S system which has agents on both client and server, and additional MCQPS which has data warehouse function of mobile client oriented. Agent is connected to server via wired network while agent communicates with mobile client via wireless network. This structure aims to lessen the communication burden by communicating via wireless network, because the mobile client has relatively low speed and limitation on carrying out large scale data processing.

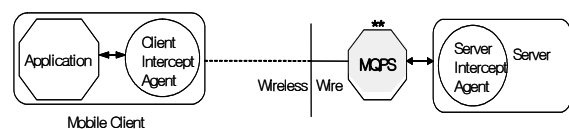


Fig. 5. New Mobile Client-Server Model with MCQPS

3.1 System components

Proposed system consists of mobile client, mobile agent and MCQPS as shown in fig.6. Mobile computing environment is made up of the mobile computers called mobile client(MC) and their wired network. Mobile client performs communication with the wired network by the computer called mobile support system(MSS).

Each MSS manages the geographical region which it is capable

to support, that is, mobile clients in the cell. And it is responsible for real-time mobile network interface which is event centered. Mobile agent can request in real-time the query with regard to the request of mobile client, and can review the result.

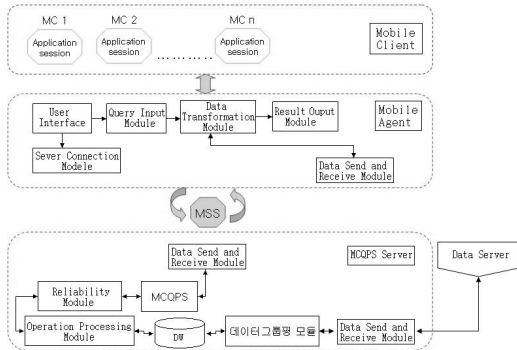


Fig. 6. New Mobile Client-Server Framework with MCQPS

Mobile client consists of user interface, server connection module, query input module, data transformation module and data send/receive module. MCQPS server consists of data send/receive module, data grouping module, data warehouse(DW), operation processing module, reliability module to maintain data synchronization and consistency and MCQPS.

3.2 MCQPS Design

3.2.1 Internal structure

When disconnection occurs, MCQPS should have the following building blocks in order to store data changes occurred at the used view and manage the complex version of view. The internal structure of MCQPS including such elements is shown in fig.7.

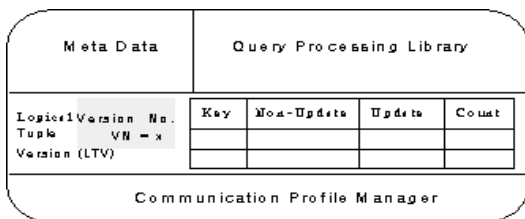


Fig. 7. Design of MCQPS Structure

3.2.2 Continuous query processing

(1) Method of optimization of near region of moving object

In case of continuous queries of moving object, the network burden caused by continuous queries can be decreased by verifying the validity of query results, and the processing of queries on particular domain can be effectively done by calculating the optimal nearest region of moving object with the consideration of dominance relationship of static property of target object. The optimal nearest region of moving object is the region where the object that dominates with regard to static property is nearer to the moving object than the object that dominates with regard to static property.

That is, optimal nearest region of is the region formed by

perpendicular bisectors which form the near region of and perpendicular bisectors created by the target objects that dominates with regard to static property. Perpendicular bisectors forming the optimal nearest region can be determined by using the near region determination method described previously.

In case the location of moving object is included, the moving object is always included in the query result as it is nearer to than the object that dominates with regard to static property, and it is also nearer to the object that dominates with regard to static property, thus this object is not included in the query results at all.

(2) Method of determination of near region of moving object

- step 1 : To calculate the objects that dominates with regard to static property, perpendicular bisectors and inequality region
- step 2 : To calculate all crossing points of perpendicular bisectors forming
- step 3 : To calculate the crossing points satisfying the simultaneous inequalities (vertices of that religion)
- step 4 : To remove the line segments which do not pass through 2 vertices

(3) Data structure of the optimal nearest region

Perpendicular bisectors forming the near region of moving object is represented in the form of line equation such that the coefficients of a, b, c in the equation, location information of for the line segments to investigate the inequality region, object id of the region and id of dominating object will be stored. In addition, to calculate the shortest distance between query location and line segment forming the region, the coordinate value of two vertices on the line, $(X_{min}, Y_{min}, X_{max}, Y_{max})$ is to be stored.

| Line ID | Dominatee ID | Dominator ID | A | B | C | Location |
|-----------|--------------|--------------|---|---|---|-----------|
| X_{min} | Y_{min} | X_{max} | | | | Y_{max} |

Fig. 8. Data Structure of Vertical Bisector

The id of the object in optimal nearest region of moving objects, id list of perpendicular bisectors forming the optimal nearest region of moving objects, id list of object encompassing the optimal nearest region for the effective investigation of the region, id list of perpendicular bisectors of near regions overlapped with the optimal nearest region are all stored.

| Object ID | Line_List | Enclosing Region List | Acrossing Line List |
|-----------|-----------|-----------------------|---------------------|
|-----------|-----------|-----------------------|---------------------|

Fig. 9. Optimal Data Structure of Nearest Region

(4) Index structure based on optimal nearest region

Indexing is performed by using the location coordinates of the target object, and the terminal node stores the location coordinates of the target object and information on the optimal

nearest region. The nearest object to the current location is searched by using the nearest neighbor query for R-tree, the near region for the moving object is calculated by using information on the optimal nearest region of the found. Fig.10 shows the R-tree configuration in which indexing is performed by using the location information of objects to be searched, based on the optimal nearest region.

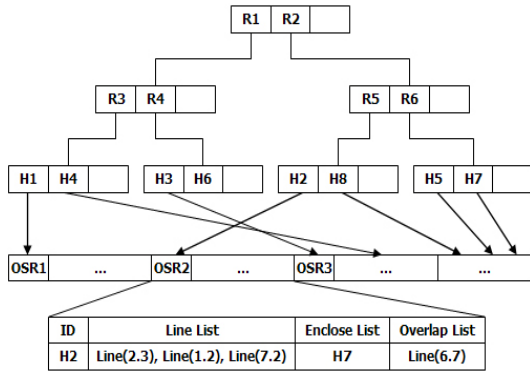


Fig. 10. Optimal R-tree Structure of Nearest Region

(5) Method of continuous queries processing based on optimal nearest region

Since the result calculated at a particular time may be invalid at other time if moving object change its location after query, the continuous queries occur as it change locations. Continuous queries incur the increased network burden and frequent recalculations. To minimize the occurrence counts of continuous queries, the distance between location coordinates of requested query and perpendicular bisectors nearest to it can be used.

The server, in response to the query request, transfers query request set, location coordinate of requested query, the distance of the nearest perpendicular bisector. The client calculates the distance between the previous location coordinate of requested query and the changed location coordinate caused by the location change, then may not request query again if the distance is smaller than the distance of the nearest perpendicular bisector, which means that the previous result cannot guarantee its validity.

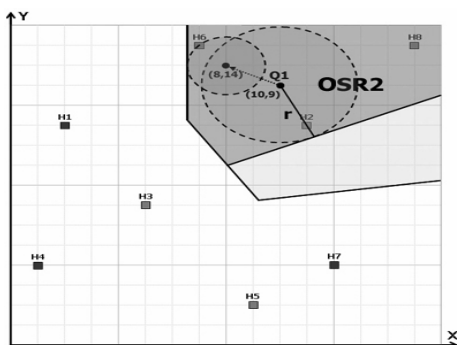


Fig. 11. Optimal Valid Region of Nearest Region

In fig.11 the server searches object H2 nearest to Q1 and

read the information of optimal nearest region of H2 by using the nearest neighbor query on R-tree. It also calculates the near region of current location by using the information of optimal nearest region of H2 and transfers query result set {H1, H2, H4, H6, H7, H8}, location coordinates(10,9) of requested query and the distance r of the nearest perpendicular bisector to the client. The client can decrease the occurrence counts of query by requesting new results only if the changed location is not included inside of the circle with radius.

3.3 Method of securing reliability

3.3.1 Three step synchronization of mobile database system

The synchronization of the existing mobile system is for the mobile database systems using some parts of data of static database, it is assumed that the data synchronization between static database and mobile database occurs periodically. Fig.12 is the schematic diagram of the overall synchronization system of the existing mobile database. The static database is one of either single or distributed database, provides mobile database with unified information. In addition, it establishes synchronization between groups by not only periodic synchronization but also using system requested agents. Each mobile group processes the necessary data and establishes the synchronization of the whole system via the query agent with the other groups where synchronization previously achieved, while maintaining its own synchronization. The synchronization of the whole mobile database system goes through the following three steps, synchronization in each group, synchronization between groups and the synchronization of central static database.

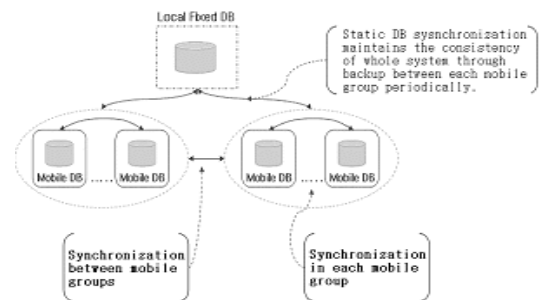


Fig. 12. Three-Level Synchronization of Mobile Database System

3.3.2 Mobile database grouping

Even though the performance of mobile database system has been improving, it has surely smaller data capacity and processing ability compared to the existing static database. Under such a circumstance, mobile database uses the selected parts of the static database by copying[5]. This means that each mobile database can distinguish the data types used by it. Based on this, we groups the each mobile database according to its respective usage. Classified mobile database is positioned in the mobile group, the policy of maintenance of data consistency and currency is applied by group.

3.3.3 Maintenance of mobile data base consistency

One of the designed functions of MCQPS is that it requires mapping the view and data source after transaction occurrence to secure the consistency of data used by MC. To do this, the following definitions (definition 1) to secure the data consistency are to be used in this paper.

(definition 1) definition to guarantee data consistency

Assume that the data source status vector $ssv_k = [ds1_k, ds2_k, \dots, ds_{m_k}]$ includes all data source statuses after performing the given transaction T_k . Then, the mapping from view to data source status m exists like this,

1. The status of each view vs_j , for any k , reflects the proper status $m(vs_j) = ssv_k$.
2. If $vs_j = vs_i$, $m(vs_j) < m(vs_i)$

3.3.4 Maintenance of mobile database currency

One of the designed functions of MCQPS is the method of recalculating the currency by creating monitoring view inter working with DBMS of server to guarantee the availability of current data of MC. To do this, it is assumed that each of data server DS1, DS2, DS3 has $r1(a,b)$, $r2(b,c)$, $r3(c,d)$ respectively, and monitoring view is created to measure the data currency signing on this table. Fig. 13 is the algorithm to determine the data currency by interworking of DMMS and monitoring view.

3.3.5 Log optimization

Under current mobile environment, the changes occurred in MC are recorded as log and this log is used when reunifying after the disconnection of MC. It is also available when frequent failures occur due to the physical and systemic reasons caused by the properties of wireless communication. The ability of managing failures to deal with such a situation is one of the essential elements to build up reliable system. In addition, minimization of log size is advantageous conditions for the saving of local memory of MC and the shortening of the time of transferring wireless band width while unifying.

4. Experiment and performance evaluation

To evaluate the performance of the indexing method by using the real-time continuous queries of moving objects and optimal nearest region, query processing method by using safe region, both of which are proposed in this paper, we carried out experiments with the simulation data of target data and moving data. To evaluate the performance of query processing method based on the near region of optimal moving object, it is experimented by comparison with the existing C-I-S model. In addition, to evaluate the performance of query processing method of effective region, we measured the number of target data and the occurrence counts of query request according to the speed of moving object.

```

public class Coordinating DBMS-agent() {
// Query DS2.r2
SELECT DS2.r2.b, DS2.r2.c ;
FROM DS2.r2 ;

B = b and C = c ;
local variable version number v = 0 ;

Start view recomputation: {
Supply query trip plan (query tree)
with Query DS2.r2 results.
Launch view evaluation DBMS-agent
with query trip plan and version number v = 0
} //end recomputation

Begin Monitoring: {
Every (Monitor_time) {
SELECT DS2.r2.b, DS2.r2.c
FROM DS2.r2

if (B  $\neq$  b or C  $\neq$  c) {
version number v = v + 1 ;
Start view recomputation: {
Supply query trip plan (query tree)
with Query DS2.r2 results ;
Launch view evaluation DBMS-agent
with query trip plan
and current version number v ;
B = b and C = c ;
} //end recomputation
} //end if
} //end Every
} //end Monitoring
}
    
```

Fig. 13. Algorithm for Data Currency Estimate

4.1 Experimentation environment

The proposed MCQPS method and method of effective region to deal with the real-time continuous queries effectively are implemented by the computer which has Pentium-IV 2.8GHz processor, main memory of 512MB and Window XP OS with 80GB hard disk, using Java language (JSDK 1.5). The data set used during the performance evaluation created 100, 200, 300 target objects and 1000 moving objects by distributing the location coordinates of both target objects and moving objects onto the 2 dimensional region with the area of 5000x 5000m through GSTD (Generator of Spatio-Temporal Datasets)[9].

To evaluate the performance of index structure and algorithm based on MCQPS propose here, we experimented by comparing it with the query processing method used in the existing C-I-S model. Table 1 shows the parameters used in the performance evaluation.

4.2 Performance evaluation

4.2.1 Experiment according to the number of target objects

To evaluate the performance of MCQPS structure proposed in this paper, the message delivery counts and the size of data transfer or message in C-I-S structure described in section 2 and in MCQPS for the service to guarantee the data reliability with regard to application session requested by MC were compared and analyzed.

Table 1. Experimentation Parameters

| Experimentation Parameters | Description | Value |
|----------------------------|---|---------------------|
| ClientNum | number of moving objects | 1000 |
| ServiceNum | number of target objects | 100, 200, 300 |
| UpperBound_Y | The maximum y value in spatial coordinate | 5000 |
| LowerBound_Y | The minimum y value in spatial coordinate | 0 |
| RightBound_X | The maximum x value in spatial coordinate | 5000 |
| LeftBound_X | The minimum x value in spatial coordinate | 0 |
| AVG_Speed | average speed of moving objects | 1, 3, 5, 10, 15, 20 |

Let's put the arbitrary data source i and data source exists in the range of $1 \leq i \leq u$. And the size of message is N , then the message size of data source Ni . On the other hand n_i represents the renewal counts of data. Now the message delivery counts of C-I-S is represented by equation (1), the size of message transfer is represented by equation (2)[8].

$$O(1 + n_i(2u-1)) \tag{1}$$

$$O(1) + n_i(O(1) + ((O(1) + O(1/N_i)))) \tag{2}$$

In the proposed MCQPS structure, data accuracy is evaluated by the algorithm suggested by fig. 13 for arbitrary message, view accuracy test for data source u was carried out. Therefore, the message delivery counts of MCQPS structure is equation (3).

$$O(u) \tag{3}$$

And, the size of transferred message or data in MCQPS structure becomes $N1+N2$ if message with the initial transfer size of $N1$ is transferred to the next site DS2. This message is transferred to DS3. Therefore, the size of transferred message in MCQPS structure is represented by equation (4).

$$N1 + (N1+N2) + (N1+N2+N3) + \dots = Nu + N_i(u-j) \tag{4}$$

As users who want to do their work by wireless network under mobile computing environment are increasing, the issues related with the weak connectivity of wireless network and with the way how to ensure the data consistency and reliability when disconnection occurs are significantly important. Therefore, the element to reduce the delivery counts of requested messages from MC and the element to minimize the size of transferred messages to overcome the limitation of mobile equipments in spite of the increased users of mobile environment are essential to provide stable services under mobile database environment.

Thus, with these two elements, we examined to what extent the optimization of message usage counts and message size in response to increasing users affect the services under wireless computing environment by comparing the performance of C-I-S and MCQPS. The result is shown at table 2.

Table 2 shows the result obtained by comparing the optimization of message usage counts and message size, according to the

request count by users, of the existing C-I-S structure with those of proposed MCQPS structure. It is proved here that the proposed MCQPS is superior to C-I-S for the stable and reliable services as users under mobile computing environment are increasing.

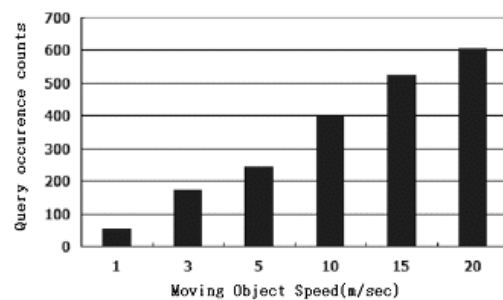
Table 2. Comparison List of Two Elements

| Data source(u) | Message delivery counts | | Size of delivered message | |
|--------------------|-------------------------|-------|---------------------------|-----------|
| | C-I-S | MCQPS | C-I-S | MCQPS |
| 1 | 2 | 1 | 3 | 1 |
| 10 | 1046 | 10 | 687 | 175 |
| 50 | 126226 | 50 | 67894 | 20875 |
| 100 | 1004951 | 100 | 523151 | 166750 |
| 500 | 125124751 | 500 | 63175600 | 20833750 |
| 1000 | 1000499501 | 1000 | 502873873 | 166667500 |

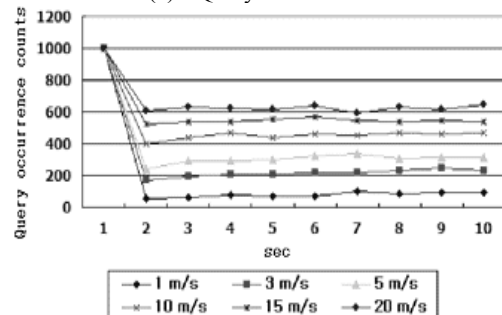
* Here, initial message size $N = 1$ / data renewal count $n = 1$ are assumed

4.2.2 Experiment according to the speed of moving object

To experiment the performance of effectiveness verifying method for the continuous queries by using the optimal nearest region based on the proposed MCQPS, we used 100 target objects distributed equally on the 2 dimensional region with the area of 5000x 5000m. We calculated and indexed the optimal nearest region for 100 target objects which have equally distributed location coordinates and the values of static property in advance, and we measured the query occurrence counts by changing the average speed of 1000 moving objects up to 1, 3, 5, 10, 15, 20m/s with Gaussian distribution. In Fig 14(a) and (b), the query occurrence counts are represented graphically.



(a) Query occurrence counts



(b) Comparison of query occurrence counts for 10 sec

Fig. 14. Query Occurrence Counts of Moving Object Speed

As shown in fig 14 (a), as the speed of moving object increases, query occurrence counts also increase because more objects exceed the effective region of the previous query. And as shown in fig 14 (b), query occurrence counts according to speed can be decreased from minimum 37.7% up to maximum 92.3%. In case that the effective region is not considered for the continuous queries, if 1000 moving objects change their location, 1000 times of new queries occur, however in case that the validity of the previous query are verified by using near region, 369 times of new queries on average occurred. In addition, as for the moving object with the relatively high speed of 20m/s, 62.4% of new queries occurred, which means reduction by 37.7%. The query occurrence counts of the whole moving objects according to the speed are reduced by average 63.2%. It is shown that the processing of continuous near region queries for moving objects by using effective region reduces the query occurrence counts caused by the location change of moving objects, which results in the network burden and recalculation cost of the server.

5. Conclusion

This paper proposed the new mobile client-server system including MCQPS to maintain the stability of data processing and data consistency under mobile computing environment which has the problem of weak connectivity and frequent disconnection, and to overcome the limitation of mobile equipment when processing large scale data.

In the existing research, the query processing method of moving objects considered only the distance to target object with dynamic properties, while near region query processing method of moving object considered only static properties with no consideration of the query location. Therefore, the continuous near region query for moving objects were not possible to be processed. This research has defined Nearest Region which is not related with the speed and direction of moving object, and suggested the method of determining the region. In addition, in order to process the query efficiently, the method of determining the Optimal Nearest Region and the method for extended index structure and query processing were proposed and implemented.

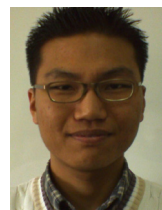
It is shown that the query occurrence counts can be decreased up to 92.3% by various experiments, which results in the reduction of the burden of network or server. The method of near region continuous query based on the MCQPS in this paper is applicable to many location based service areas.

References

- [1] H. J. Joo, Y. B. Park, "Design of Reliable Query Processing System in Mobile Database Environments," *The KIPS Transactions Part D*, pp. 521-530, 2005.
- [2] J. W. Lee, "Application Case of Mobile Database," *Journal of*

Database Research, vol. 17, pp. 115-118, 2001.

- [3] M. S. Choi and Y. K. Kim, "Introduction to Mobile Database," *Journal of Database Research*, vol. 17, pp. 3-16, 2001.
- [4] Margaret H. Dunham and Vijay Kummer, "Impact of Mobility on Transaction Management," *Proceeding of the International Workshop on Data Engineering for Wireless and Mobile Access*, pp.14~21, August, 1999.
- [5] H. Joanne and A. Divyakant, "Planned Disconnections for Mobile Database," *Proceedings of IEEE 11th international workshop*, 2000.
- [6] Sanjay Kummur Madria, Bharat K. Bhargava, "A Transaction Model to Improve Data Availability in Mobile Computing," *Distributed & Parallel Databases*, vol. 10, no. 2, pp. 127 -160, 2001.
- [7] G. Walborn and P. K. Chrysanthis, "Proceeding in Mobile Database Applications," *In Proceeding of the 14th Symposium on the Reliable Distributed Systems*, September, 1995.
- [8] S. W. Lauzac, "Utilizing Customized Materialized Views to Create Database Services Suitable for Mobile Database Applications," Ph. D thesis, Pittsburgh Univ., 2001.
- [9] Y. Theodoridis, J. R.O. Silva, and Mario A. Nascimento, "On the Generation of Spatiotemporal Datasets," *In Proceedings of the 6th Int'l Symposium on Large Spatial Database(SSD)*, 1999.
- [10] B. Zheng and D. Lee, "Semantic Caching in Location-Dependent Query Processing," *SSTD*, pp. 97-116, 2001.
- [11] Z. Song and N. Roussopoulos, "K-Nearest Neighbor Search for Moving Query Point," *SSTD*, 2001.
- [12] S. Borzsonyi, D. Kossmann and K. Stocker, "The Skyline Operator," *In ICDE*, pp. 421-430, 2001.
- [13] K. Tan, Eng, and B. Ooi, "Efficient Progressive Skyline Computation," *In VLDB*, pp. 301-310, 2001.
- [14] D. Kossmann, F. Ramsak, and S. Rost, "Shooting Stars in the Sky: an Online Algorithm for Skyline Queries," *In VLDB*, pp. 275-286, 2002.



Young-Baek Kim received the B.S. and M.S. degrees in computer science and engineering from Kyungnam University, Changwon, Korea in 2005 and 2007, respectively. He is currently in the Ph.D. course. His current research interests include image processing, context awareness and

augmented reality.

Phone : 055 -249 -2706

E-mail : baroaleum@gmail.com



Young-Choon Kim received the B.S in electrical engineering from Daejeon Engineering College in 1987 and the M.S. and Ph. D. in electrical engineering from Myungji University, Seoul, Korea in 1989 and 1997, respectively. He is currently a Full Professor in the Division of Mechanics and Car Engineering, Kongju National University, Korea. His current research interests include electrical power conversion of electric automobile, automotive electronics control, hybrid car.

Phone : 041 - 521 - 9274
E-mail : yckim59@kongju.ac.kr



Hae-Jong Joo received the Ph. D. of Education from Cumberland University, USA in 2008 and the Ph. D. in Computer Engineering from Myungji University, Seoul, Korea in 2010. He was a assistant professor in department of multimedia, Daewon College and a professor in the department of hunic, Dongguk University, Korea. His current research interests include data engineering, IT conversion technologies, ubiquitous business.

Phone : 070 - 8680 - 7006
E-mail : hjjoo@dongguk.edu



Sang-Yong Rhee received the B.S. and M.S. degrees in industrial engineering, Korea University, Seoul, Korea in 1982 and 1984, respectively and the Ph.D. degree in industrial engineering, Postech, Pohang, Korea in 1992. He is currently a Full Professor in the Division of Computer Science and Engineering, Kyungnam University, Korea. His current research interests include computer vision, neuro-fuzzy system, intelligent robot and augmented reality.

Phone : 055 -249 -2706
E-mail : syrhee@kyungnam.ac.kr