Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X (P) Vol. 3 No. 1, 2014, pp. 50-56

DOI: https://doi.org/10.51983/ajeat-2014.3.1.693

Location Detection and Time Monitoring System for Moving Object

Nirupama.Sheelwant[1] and Sandeep.Sheelwant[2]

[1]*Department of CSE, Lingaraj Appa Engg College, Bidar, VTU, Karnataka*

[2]*Department of EEE, PIRMEC, Hydrabad, JNTU, A.P*

E-mail: ....

(Received on 25 March 2014 and accepted on 18 June 2014)

Abstract - **Location tracking and updating are the fundamental issues in the moving objects. Object's moving efficiency and location detection accuracy are taken into account in the project. The client sends the queries to the server based on the proper result. The server sends the reply to the client maintaining its accuracy as well as efficiency. Main strategies are to update the client request with the exact result. The advantage of this is the flexibility and ability to optimize the efficiency and also updating location of the moving object. We developed a system of high efficiency and accuracy to detect the exact location and also location updating technique for moving object.**

Keywords : **Moving Object, Location update, Query Index, Safe Regions, Object index.**

I. Introduction

In mobile and spatiotemporal databases, monitoring continuous spatial queries over moving objects is needed in numerous applications such as public transportation, logistics and location-based services [1].

Monitoring system consists of a base station, database server, application servers, and a large number of moving objects (i.e., mobile clients). The database server manages the location information of the objects. The application server gather monitoring requests and register spatial queries at the database server, which then continuously updates the query results until the queries are deregistered.

The expanding use of location-based services [2] has profound implications on the privacy of personal information. If no adequate protection is adopted, information about movements of specific objects could be disclosed to unauthorized subjects or organizations, thus resulting in privacy breaches and hence, there is a framework for preserving location privacy in moving-object environments.

The continuous query monitoring is focused on location updates. Two commonly used updating approaches are periodic update (every client reports its new location at a fixed interval) and deviation update (a client performs an update when its location or velocity changes significantly) [3] [4]. However, these approaches have several deficiencies. First, as the monitoring accuracy is low, query results are correct only at the time instances of periodic updates, but not in between them or at any time of deviation updates. Second, as location updates are performed regardless of the existence of queries–a high update frequency may improve the monitoring accuracy, but is at the cost of unnecessary updates and query revaluation. Third, the server workload using periodic update is not balanced over time; hence it reaches the peak when updates arrive and trigger query revaluation, but is idle for the rest of the time. Last, the privacy issue is simply ignored by assuming that the clients are always willing to provide their exact positions to the server. The number of objects is some orders of magnitude larger than that of queries. As such, the query index can accommodate all registered queries in main memory, while the object index can only accommodate all moving objects in secondary memory. This assumption has been widely adopted in many existing proposals [5] [6]. The database server handles location updates sequentially in other words; updates are queued and handled on a first-come-first- serve basis. This is a reasonable assumption that eliminates issues of read/ write consistency.

The moving objects have connection with the database server. Furthermore, the communication cost for any location update is a constant. With the latter assumption, minimizing the cost of location updates is equivalent to minimizing the total number of updates.

II. System Framework

A. ITS (Intelligent Transport System) Framework

The system framework comprises of an application server which is connected to a Mobile Client, database and transport server that is used instead of GPRS. Mobile clients and transport server interact through application server.

Mobile clients have to register themselves with the application server. When mobile clients send request queries, all the queries are forwarded to application server and then it is given to transport server for processing. Transport server sends the result to application server which in turn sends to Mobile clients. There is a map that shows the moving objects which are nothing but buses along with the requested information.

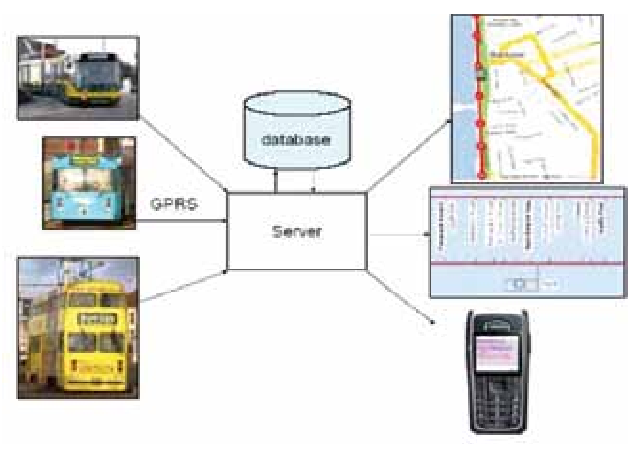

Fig 1. ITS Framework

B. System Architecture

Large systems are always decomposed into sub-systems that provide some related set of services. The initial design process of identifying these sub-systems and establishing a framework for sub-system control and communication is called architecture design and the output of this design process is a description of the software architecture.

The architectural design process is concerned with establishing a basic structural framework for a system. It involves identifying the major components of the system and communications between these components. The system architecture shows the blocks required for the project. Fig.2 shows the existing system architecture

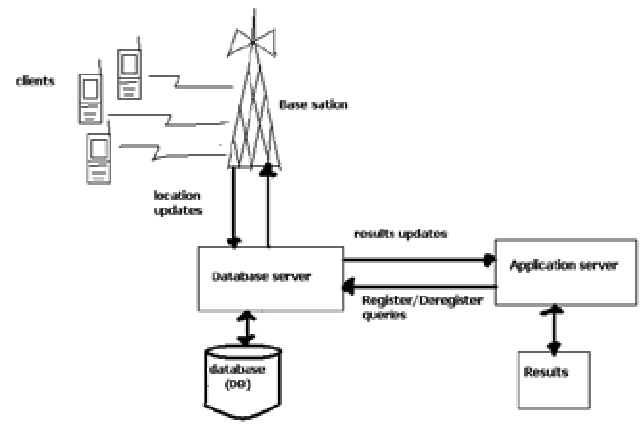

Fig 2. System Architecture

III. Related work

A monitoring framework [7] was proposed where the clients were aware of the spatial queries being monitored, so they could send location updates only when the results for some queries might change. Basic idea there was to maintain a rectangular area, called safe region, for each object. The safe region is computed based on the queries in such a way that the current results of all queries remain valid as long as all objects reside inside their respective safe regions.

A client updates its location on the server only when the client moves out of its safe region. This significantly improves the monitoring efficiency and accuracy compared to the periodic or deviation update methods. However, this framework fails to address the privacy issue, that is, it only addresses when but not how the location updates are sent and also earlier work assumed a static data set and focused on efficient access methods [8] and query evaluation algorithms [9] [10]. Recently, a lot of attention has been paid to moving-object databases, where data objects or queries or both of them move.

Some recent work attempted to remedy the privacy issue. Location cloaking was proposed to blur the exact client positions into bounding boxes [11] [12] [13] [14]. By assuming a centralized and trustworthy third-party server that store all exact client positions, various location cloaking algorithms were proposed to build the bounding boxes while achieving the privacy measure such as k-anonymity. However, the use of bounding boxes makes the query results no longer unique. As such, query evaluation in such uncertain space is more complicated. A common approach is to assume that the probability distribution of the exact client location in the bounding box is known and well formed. Therefore, the results are defined as the set of all possible results together with their probabilities [15].

However, all these approaches focused on onetime cloaking or query evaluation; they cannot be applied to monitoring applications where continuous location update is required and efficiency is a critical concern.

A more comprehensive approach is introduced instead of dealing with when and how separately like most existing work, a ITS framework has been proposed which incorporates the accuracy, efficiency, and privacy issues altogether. However, the integration of privacy into the monitoring framework poses.

Challenges to the design of ITS system:

- First, with the introduction of bounding boxes, the result of a query is no longer unique.
- Second, the most probable result also adds complexity to the definition of safe region. New algorithms must be designed to compute maximum safe regions in order to reduce the number of location updates, and thus, improve efficiency.
- Third, as the location updater decides when and how a bounding box is updated,

Its strategy determines the accuracy, privacy, and efficiency of the framework. The standard strategy is to update when the centric of the bounding box moves out of the safe region, which guarantees accuracy no miss of any change of the most probable result. To optimize privacy or efficiency, however, alternative strategies must be devised.

Compared to the previous work [7], the ITS framework that addresses the issue of location updating holistically with monitoring accuracy, efficiency, and privacy al together. This framework extends from our previous work [7] by introducing a Common privacy model, and therefore, suits realistic scenarios.

Limitation in Existing System

There is no information provided to the passengers regarding the time duration of the vehicle to reach the passenger, Details.

Existing System Features

i. Bus Schedule, displayed at the bus terminal.

ii. Details of buses and route information, with no other information such as time.

Deficiency of the Existing System

Provides no details about the location of the current bus for a route involves indefinite waiting at the terminal for an unknown location of the particular bus and its route.

IV. Algorithmic Approaches

Step 1: Application server can send register queries to database server.

Step 2: When an object send the location updates to server.

Step 3: the query processors are identified those register queries that affected by update using query index and then revaluate them using object index.

Step 4: Server is updating query result and send back result to application server who register these queries.

Step 5: Application server updating based on location of object and send response to client /mobile terminal.

V. Model workflow

5.1 Data Flow Diagram

Data Flow Diagram (**DFD**) is a graphical representation of the “flow” of data through an information system. Data Flow models are used to show how data flows through a sequence of processing about the next nearest vehicle available, Distance between the vehicle and the passenger. It is mere the passenger has to stand in the stop acknowledging the bus time table while waiting indefinitely, until the next vehicle comes to pick them up. steps. The data is transformed at each step before moving on to the next stage. These processing steps or transformations are program functions when Data Flow diagrams are used to document a software design.

The Data Flow Diagram (DFD) for the privacy-aware monitoring Application is divided into three levels such as level 0, level 1 and level 2.

Safe Region Evaluation

In this safe region is assumed as a rectangle change of object inside the rectangle would not affect spatial query in the database. The safe region is computed based on the queries in such a way that the current results of all queries remain valid as long as all objects reside inside their respective safe regions. The safe region computing is based on the rectangle of the centric.

Object index

It is the server side information about spatial query range and used to evaluate safe region.

Query Index

It has the parameters as query point, current query result and the quarantine area. The quarantine area is used to identify the queries whose results might be affected by an incoming location update. The number of objects is some orders of magnitude larger than that of queries. As such, the query index can accommodate all registered queries in main memory, while the object index can only accommodate all moving objects in secondary memory.

Query Processing

In the System framework, based on the object index, the query processor evaluates the most probable result when a new query is registered, or revaluates the most probable result when a query is affected by location updates. Obviously, the revaluation is more efficient as it can be based on previous results.

Location Updater

The each time a client detects the genuine point location, it is wrapped into a bounding box. Then, the client-side location updater decides whether or not to update that box to the server without any other knowledge about the client locations or moving patterns, upon receiving such a box, the server can only presume that the genuine point location is distributed uniformly in this box.

Data Flow Diagram – Level 0

The level 0 is the initial level Data Flow Diagram. It is common practice for a designer to draw a context-level DFD first which shows the interaction between the system and outside entities.

This context-level DFD is then “exploded” to show more detail of the system being modelled.

Fig 5.1(a) shows the Level 0 Data Flow Diagram.

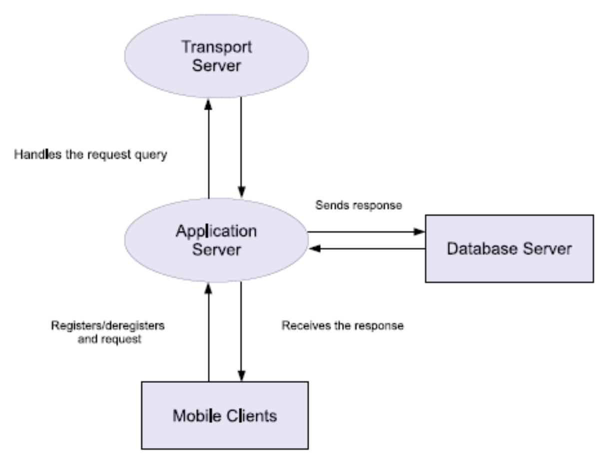

Fig 5.1(a) Level 0 DFD

Data Flow Diagram – Level 1

The Level 1 Data Flow Diagram gives more information than the level 0

Data Flow Diagram. Fig 5.1(b) shows the Level 1 Data Flow Diagram.

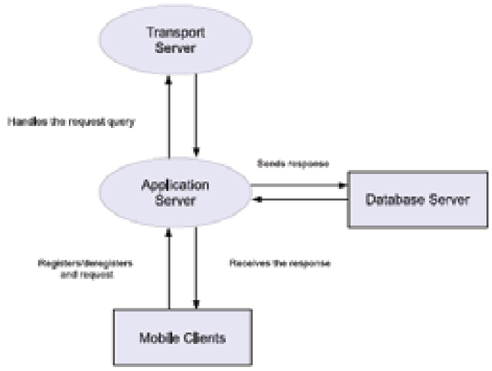

Fig 5.1(b) Level 1 DFD

Here, client's updated request query processing is done by the transport server based on the query index and the processed result is fetched by the application server using object index.

Data Flow Diagram – Level 2

The Level 2 Data Flow Diagram gives more information than the level 1.

Data Flow Diagram. Fig 5.1(c) shows the Level 2 Data Flow Diagram.

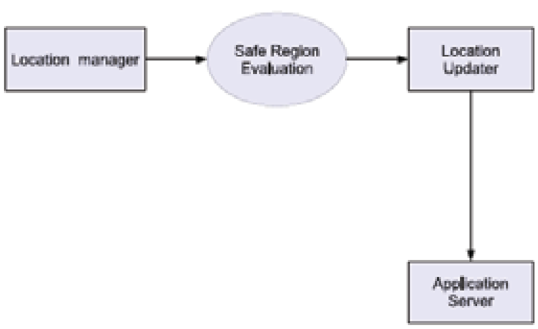

Fig 5.1(c) Level 2 DFD

Here, location manager takes care of evaluation of safe region which is a rectangular area considered for the moving objects, updates the location of moving objects and sends the result to the application server.

5.2 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Hence they can be regarded as a form of flowchart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

There are four main modules in this project.

- Server Module
- Client Module (Mobile Terminal)
- Object Location Application Module
- Database & Mobile Terminal Connectivity Module.

The sequence of data flow and control flow among each module/function is shown in the Activity Diagram Fig.5.2 When mobile client sends a request query, it is forwarded to transport server via application server. Transport server processes the query and sends the result back to mobile client via application server.

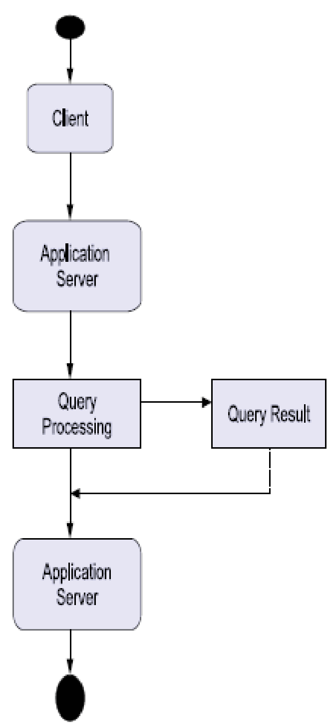

Fig 5.2.Activity Diagram

VI. Experiment Setup

To evaluate the monitoring performance of moving object, we implemented this framework to simulate the proposed work that how a server dynamically monitors the moving objects and a server to provide the updating result of object in each second and continuously monitoring a moving object. All details are stored into database.

Fig 6 (a) shows the client module which allows the user to get register with the application server by entering the valid username, password and IP Address of the local web server.

Fig 6(a) Client registration with the Application server

Fig 6(c) Client module that shows the selection of source

Fig 6(d) shows the client part showing the requested vehicle information such as nearest bus, bus location, time to reach source and the destination up on selecting the source and destination.

Fig 6(b): Transport Server Map with Moving Object

Fig 6(b) shows the Transport server map containing three buses. There is start button to start each of these buses and when the bus starts its details are displayed on the right side of map. This shows the details of Bus1 on the map up on selecting the start Bus1 button.

Fig 6(c) shows the client part after the registration process where the user has to select the source first and then destination in order to get the requested information about the vehicle.

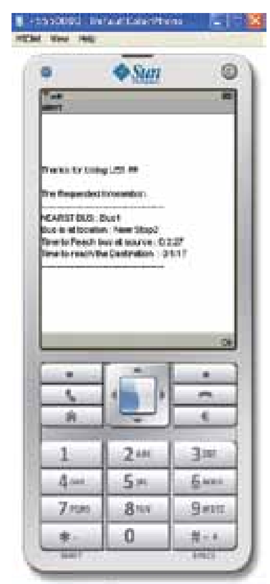

Fig 6(d): Client module that shows the requested information

VII. Result Analysis

For the above experiment setup, we executed the ITS framework and thus shown the tracking or monitoring of the moving object & location update depends upon the user choice. There are few challenges that if user wants to know the information regarding moving object, they have to register and deregister every time.

VIII. Conclusion and Future Scope

This project focuses on providing an economical solution for public transport which is made available to people at different levels. The integrated architecture follows the trend of Intelligent Transportation System (ITS). Since the third party service provider is provided by using the mobile phone network, the operational cost can be reduced, giving a successful economical real-time solution. Some users may need information regarding the bus timings like details about the departure point of the bus, nearest bus number, information about its arrival point, ITS Framework is the first to address the issue of location updating with regard to monitoring accuracy, efficiency, and privacy. Two-client update strategies that optimize accuracy and efficiency are used. Furthermore, the framework is robust and scales well with various parameter settings, such as privacy requirement, moving speed, and the number of queries and moving objects.

In our Future Enhancements are Varied and flexible approaches could be used to provide a commuter with real-time data about a particular bus service. The application could be enhanced to include provision for monitoring the time delay due to the traffic scenario for the given bus route so as to give the exact time and distance of the required bus. The application could be enhanced to monitor the vehicles based on the traffic scenario. Other types of queries can be incorporated into the framework, such as spatial joinsand aggregate queries. In particular, the minimum cost update strategy shows that the safe region is a crude approximation of the ideal safe area, mainly because the safe region is separately optimized for each query, but not globally. A possible solution is to sequentially optimize the queries but maintain the safe region accumulated by the queries optimized so far. Then, the optimal safe region for each query should depend not only on the query, but also on the accumulated safe region.

References

[1] Haibo Hu, Jianliang Xu, and Dik Lun Lee, "PAM: An Efficient and Privacy- Aware Monitoring Framework for Continuously Moving Object ", IEEE transactions on knowledge and data engineering, march 2010.

[2] Dan Lin, Elisa Bertino, Reynold Cheng, Sunil Prabhakar,"Location Privacy in Moving-Object Environments", Transactions on Data Privacy , 2009.

[3] C.S. Jensen, D. Lin, and B.C. Ooi,"Query and Update Efficient B+-Tree Based Indexing of Moving Objects", Intl Conf. Very Large Data Bases (VLDB), 2004.

[4] M.F. Mokbel, X. Xiong, and W.G. Aref, "SINA: Scalable Incremental Processing of Continuous Queries in Spatio-Temporal Databases", Proc. ACM SIGMOD, 2004.

[5] D.V. Kalashnikov, S. Prabhakar, and S.E. Hambrusch, "Main Memory Evaluation of Monitoring Queries over Moving Objects", Distributed Parallel Databases,2004.

[6] X. Yu, K.Q. Pu, and N. Koudas, "Monitoring k-Nearest Neighbor Queries over Moving Objects", Proc. IEEE Intl Conf. Data Eng. (ICDE), 2005.

[7] H. Hu, J. Xu, and D.L. Lee, "A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects", Proc. ACM SIGMOD, 2005.

[8] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching", Proc ACM SIGMOD,1984.

[9] G.R. Hjaltason and H. Samet, "Distance Browsing in Spatial Databases", *ACM Trans. Database Systems,*265-318, 1999.

[10] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest Neighbor Queries", Proc.ACM SIGMOD, 1995.

[11] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking", Proc. MobiSys, 2003.

[12] B. Gedik and L. Liu, "Location Privacy in Mobile Systems: A Personalized Anonymization Model", Proc. IEEE Intl Conf. Distributed Computing System (ICDCS), 2005.

[13] M.F. Mokbel, C.Y. Chow, and W.G. Aref, "The New Casper: Query Processing for Location Services without Compromising Privacy", Proc. Intl Conf. Very Large Data Bases (VLDB), 2006.

[14] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing Location- Based Identity Inference in Anonymous Spatial Queries", *IEEE Trans. Knowledge and Data Eng,* Dec. 2007.

[15] J. Chen and R. Cheng, "Efficient Evaluation of Imprecise Location-Dependent Queries", Proc. IEEE Intl Conf. Data Eng. (ICDE), 2007.