

Position Information Storage System based on RDBMS

In-Sung Jang, Dae-Soo Cho, Jong-Hyun Park

Spatial Information Technical Center, ETRI
161 Gajeong-dong Yuseong-gu, Daejeon 305-350, Korea
{e4dol2, junest, jhp}@etri.re.kr

Abstract: Recently, owing to the rapid progress of Telecommunication technology, the increase of wireless internet's subscriber and diffusion of wireless device, LBS (Location Based Services) which take advantage of user's location information and receive information in concerning with user's location to be essential services. Location Based Services are related the moving objects which change their locations through time. Therefore, to provide location-based services efficiently, it is required that an efficient system which could acquire, store, and query the large number of locations. In this paper, we design management system to insert and search a huge number of Moving Object based on Legacy Relational Database.

Keywords: LBS, MOVING OBJECT, STORAGE.

1. Introduction

These days, owing to the rapid progress of Telecommunication technology, diffusion of wireless internet and performance elevation of mobile device, Location-based Services that are based on user's location information have been useful service. Although there is some difference in definition of LBS, it commonly known as that it employ accurate, real-time positioning to connect users to nearby points of interest, advise them of current conditions such as traffic and weather, or provide routing and tracking information - all via wireless devices (ex : portable phone, PDA, notebook PC and etc)[1]

To provide Location-Based Service that takes advantage of feature that is mobility to user, we need Moving Object Management System(MOMS) [2, 3, 4] that can efficiently manage user's location information that be changed continuously.

The rest of this paper is organized as follows. In section 2, we introduce related work with moving objects. In section 3, we describe system architecture of MOMS (Moving Object Management System). In section 4, we shall show two storage module, which simple location storage module and distributed location storage module to insert and search efficiently location data in large amount to legacy database. Lastly, in section 5 we offer conclusions.

2. Related Works

Moving objects are that their state in space changes over time. As computing power and technology grows, new advanced applications manage moving objects, such as land parcel, roads, taxis, buses, fishing boats, air

planes, cars, and cellular phone users, etc. During last a decade, research about spatiotemporal databases has been a active research field.

Guting have developed a data type oriented approach for moving objects[5]. The idea is to consider the two major abstractions moving point and moving region as abstract data types like Fig1.

The group of Wolfson has proposed a concept of moving objects databases that is complementary to Guting[6]. Whereas Guting's approach of modeling describes movement in the past, hence the complete history of moving objects, their focus is on capturing the current movement of entities, e.g. vehicles, and their anticipated locations in the near future. The basic idea is to store in a database not the actual location of an moving object, which would need to be updated frequently, but instead a motion vector describing location, velocity and direction for a recent instant of time. As long as the predicted position based on the motion vector does not deviate from the actual position more than some threshold, no update to the database is necessary.

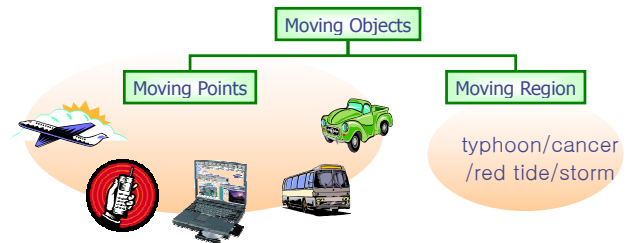


Fig. 1. Moving Object.

3. Moving Object Management System

In this section, we describe Moving Object Management System (MOMS).

MOMS consists of three major components, which are Query Processor Component, Storage Component, Index Component like Figure 2. And related modules are gateway and Application. Through various location acquisition strategies, Gateway acquires current location of moving object. It is gotten by network based moving object, handset-based object such as GPS from SKT, KTF and LGT. In this paper, we use location information of moving object generated by GSTD, City Simulator for test

In brief, the function of each component is same as following. Location Query Component is that executes query based on model of moving object and its operator. The Index Component maintains two indexes at the same time. One is current location index, and the other is past location index. Current Location Index takes only current locations of continuously moving objects into consideration. Past Location Index has a special purpose of efficient processing of a time interval queries and a trajectory queries. Location Storage Component is to store moving object reported from gateway and to search moving object that correspond to query result of Location Query Component. Examine particularity in Section 4.

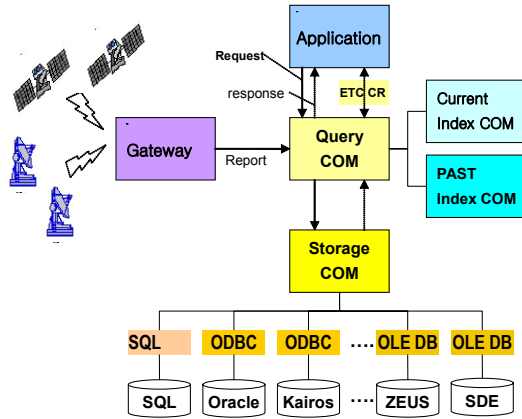


Fig. 2. Moving Object Management System.

4. Location Storage Module

1) Simple Storage Module

Simple Storage Module (SSM) is to do insert and search by Query Component. For this, SSM manages location information by using Dot NET Data Provider. Dot NET Data Provider supports three types, which are SQL Client, OleDb and ODBC. Therefore, if databases supported .Net Data Provider, SSM can use by same interface heterogeneous database system like Fig 3.

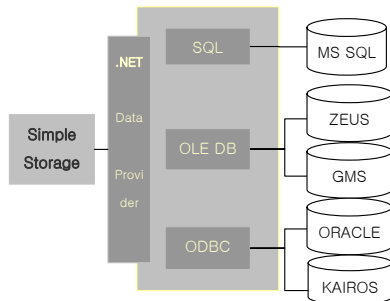


Fig. 3. .Net Data Provider

It describes table schema at Table1 and Table2. Table1 shows immediately update. On inserting Location information, which consists of position coordinates, acquisition time and error by Query Component. It is very

simple architecture, but it has a few defects. Above all, it has a number of transactions. Therefore it have load in MODB which have frequently update. Next, it is difficult to analyze trajectory of Moving Object. Next, it has duplicate **MOID**.

Table 1. Example of Simple Table and Schema

Field	Type	Example	Desc
MOID	Long	0123456789	Moving Object ID
X	Double	38492.32	X Coord
Y	Double	56431.52	Y Coord
MOTime	DateTime	"2003-09-15 11:02:32"	Acquired Time
MOError	Float	100	Error Range

Table 2. Example of composite table and Schema

Field		Type	Example	Desc
MOID		Long	0123456789	Moving Object ID
M B R	MinX	Double	31000.00	Min X Coord
	MaxX	Double	55000.00	Max X Coord
	MinY	Double	41000.00	Min Y Coord
	MaxY	Double	66431.52	Max Y Coord
FromTime		DateTime	"2003-09-15 11:02:32"	Start Time
ToTime		DateTime	"2003-09-16 12:02:32"	End Time
Data Length		UInt	20	Location Number
Locations		BLOB or String	0XF23F3D123CB5.. "31001,35412.9,20,"	Locations

So we design buffer to group locations of a moving object by **MOID**. Data Structure of buffer is hash table or B-tree. Key value is **MOID**, and Data value consists of **Length** (the number of locations stored in it), **MBR** (Minimum Bounding Rectangle of the locations), **From** (time that first location in it is acquired), **To** (time that last location in it is acquired) and **Locations** which represent a trajectory of a moving object from **From** time to **To** time. In considering performance, we prefer to hash Table. Fig 3 shows the schema of composite table for this. At this, type of locations columns is BLOB (Binary Large Object) or string, if storage system do support BLOB type or not.

2) Distributed Storage Module

The characteristic of MODB is that heavy transactions been at frequently interval. To solve this traffic, we suggest Distributed Storage Module (DSM). DSM consists of Remote Server, Server Manager and Distributed Meta Manager. Each subcomponent is following. The role of Server Manager(SM) maintains information of Remote Server (RS). It has function of registering RS, un-registering RS and checking RS statues. And it keeps

load balancing in inserting new locations. For this, SM maintains weight table like Fig 4. In here, W_i means weight for Remote Server's performance which is reflected in network traffic and CPU power.

$$W_i = \frac{n}{\text{the time required to insert } n \text{ - Rows into DB}_i \text{ (RS)}}$$

For example, Fig 4 shows each database weight and procedure. Let's each database weight is 4:2:1. In first inserting, SM selects DB₁ because its weight is high. Then, SM update weight table by 3:2:1. In 2nd inserting, it selects DB₁ because its weight is high as before. In 3rd inserting, if weight is equal, it select database using LRU (least recently used) algorithm. So SM selects DB₂. Next step is the same.

Weight Table				
W1	w2	w3		
4	2	1		DB1 Weight = 4
After	DB1	DB2	DB3	
origin	4	2	1	
1 st	3	2	1	DB2 Weight = 2
2 nd	2	2	1	
3 rd	2	1	1	
4 th	1	1	1	
5 th	1	1	0	DB3 Weight = 1
6 th	1	0	0	
7 th	0	0	0	
8 th	4	2	1	

Fig. 4. Weight Table

Fig 5 shows the step of inserting Moving Objects. First, DSM must register RS into SM. Next, Client requests SM to select RS. Then, SM responses to Client which RS is selected using weight table. Next, Client inserts location information to correspond to RS for step 3 in fig 5. Lastly, after client inserts locations into database system through RS, client inserts meta-info into DMM (Distributed Meta Manager) which controls distributed-indexing database that contains where locations is, in distributed environment.

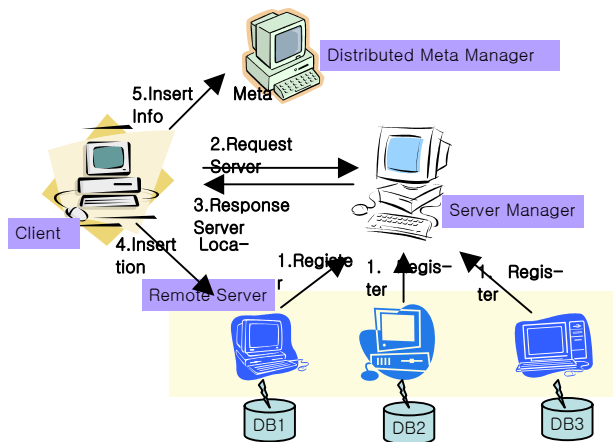


Fig. 5. Step of insert for Distributed Storage

And, Fig 6 shows the case of searching for query. DMM returns list of relevant RS information to search for query. And then, according to the RS list, Client retrieves RS.

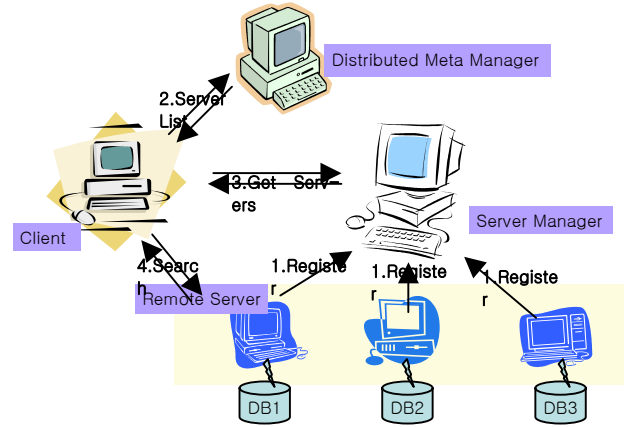


Fig. 6. Step of Searching for Distributed Storage

5. Conclusions

Recently, due to an explosive increase of interest in LBS, it is required that an efficient system which could acquire, store, and query the large number of locations. This case, location information varies from hour to hour. Addition to that, it is very huge size. For this, in this paper, we proposed moving objects storage system based RDBMS. We design two types which SSM and DDM can store and search current and past location information effectively to a diverse set of database systems. As future work, we should develop algorithms to enhance the performance of location storage.

REFERENCE

- [1] SoftBank Research, IT Insight Strategy Report, "LBS, Now & Future"
- [2] Ouri Wolfson, Bo Xu, Sam Chamberlain, and Liqin Jiang, "Moving Objects Databases: Issues and Solutions," SSDBM 1998, 111-122
- [3] Ouri Wolfson, Sam Chamberlain, Son Dai and Liqin Jiang, "Location Management in Moving Objects Databases" WOSBIS 1997, 7-12
- [4] Jensen, C.S, Jensen, A. Friis-Christensen, T.B Pdersen, D. Pfoer, S. Saltenis, and N. Tryfona, "Location-Based Services – A DataBase Perspective," Proceedings of the Eighth Scandinavian Research Conference on Geographical Information Science, As, Norway, June 25-27, 2001, pp 59-68.
- [5] Erwig, M., Guting, R. H., Schneider, M., and Vazirgiannis, M., "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Obejct in Databases," GeoInformatica, Vol.3, No.3, pp.269-296, 1999.
- [6] Sistla, A. P., Wolfson, O., Chamberlain, S., and Dao, S., "Modeling and Querying Moving Obejcts," ICDE, pp.422-432, 1997.