LOCATION INFORMATION STORAGE SYSTEM BASED ON FILE

I. S. Jang, D.S. Cho
LBS Research Team, Telematics Research Division, ETRI, 161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350 Korea

(e4dol2,junest)@etri.re.kr

Commission PS, WG IV/4

KEY WORDS: Moving Object, GPS, LBS, Storage, Database

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 File

1. INTRODUCTION

These days, owing to the rapid progress of Telecommunica-tion 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)

To provide Location-Based Service that takes advantage of feature that is mobility to user, we need Moving Object Management System(MOMS) (Sistla et al., 1997; Forlizzi et al., 2000; Wolfson et al. 1998; Güting et al., 2000)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 design Location Storage Component to insert and search efficiently location data in large amount. And in section 5, we shall show implementation on the system. Lastly, in section 6 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(Güting et al., 2000) . The idea is to consider the two major abstractions moving point and moving region as abstract data types like Figure 1.

The group of Wolfson has proposed a concept of moving objects databases that is complementary to Guting(Sistla et al., 1997).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.

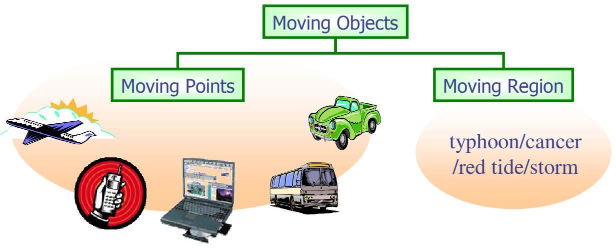


Figure 1. Moving Object.

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, Location 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.

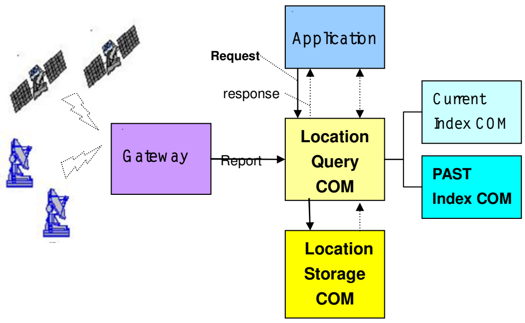


Figure 2. Moving Object Management System.

3.1 Current Location Information Component

Role of CLIC(Current Location Index Component) manage current location information and index about it (that is acquired most recently). CLIC supports spatial-based index on location and object-based index on MOID (Moving Object Identification) like Figure 3. By using Object-based index, we can confirm location information of given MOID. And, by using Spatial-based index, we can do Moving Object correspond to given spatial query. Then, Spatial-based index and Object-based index reside in main memory to update frequent location information and to execute query efficiently.

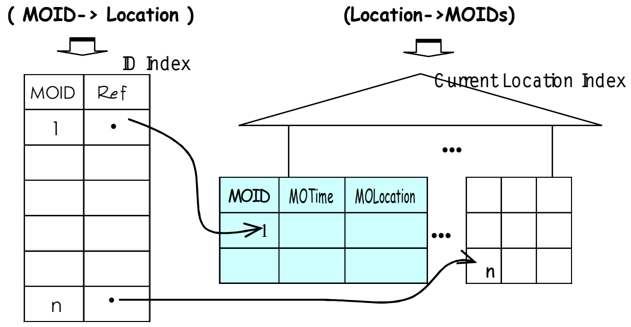


Figure 3. Current Location Index

3.2 Past Location Information Component

PLIC (Past Location Information Component) manages past location information and spatiotemporal-index about it. PLIC store location data and retrieve it by using that index. If it establishes spatial filter and temporal filter and executes query, then PLIM return set of MOID of relevant objects like Figure 4. This component must be able to run trace of wheels query about past location information of Moving Object effectively.

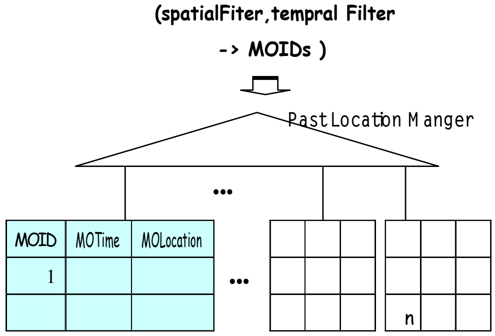


Figure 4. Past Location Index

4. THE DESING OF SYSTEM

The main function of Location Storage Component is to insert moving object that is reported from Gateway and to search moving object that correspond to query of Location Query Component.

4.1 System Architecture

The Architecture of File-Based Location Storage System is like Figure 5.

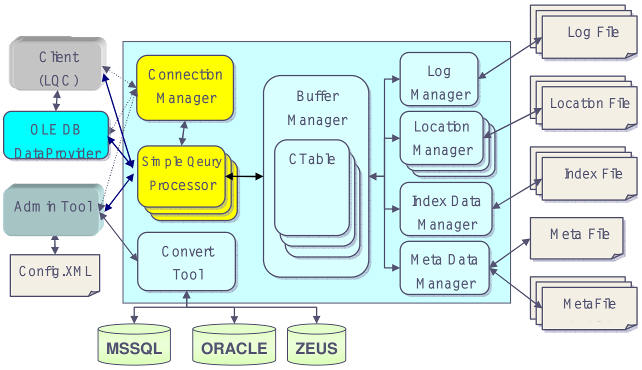


Figure 5. The Architecture of File-Based Location Storage System

The System consists of Connection Manager, Simple Query Processor, Buffer Manager, Log Manager, Location Manager, Index Manager, Meta Manager, Admin-Tool, Convert-Tool and OLE-DB Data Provider. Each Module is shown next.

4.2 The Structure of Directory

The structure of system directory is like Figure5. System is correspondence to home, and each database is correspondence to directory. Tables in Database are correspondence to subdirectories of database directory. In each database directory, there are log directory which have log file and mete directory which have meta-data of database and meta-data of table. And there are zero and more table directory. In table directory, there is data file and index file.

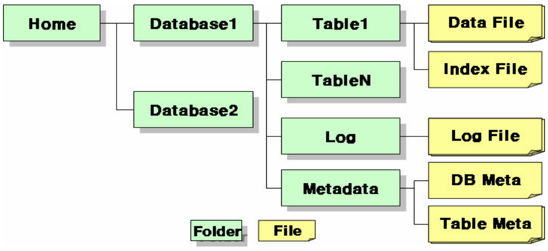


Figure 6. Directory Structure

4.3 MORow Object and MOPage Object

In Figure 7, it shows the structure of MORow object. It is composed of MOID (Moving Object IDentifier), Length (the number of locations stored in it), MBR (Minimum Bounding Rectangle of the locations), From (time that first location in it is acquired) and To (time that last location in it is acquired). Therefore, the MORow object represents a trajectory of a moving object from From time to To time.

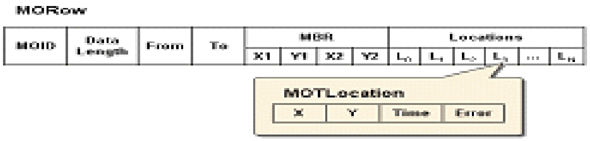


Figure 7. MORow Structure

The MOPage is similar to MORow. That is file structure to store locations to file, just as MORow is memory structure. It shows the structure of MOPage object in Table1 1. if the DataLength of a MORow is equal to the MaxDataLength, then all of the locations in the MORow object to be store into physical file permanently by MOPage. MaxDataLength is estimated by PageSize.

Table 1. MOPage Structure

CurrentPageLink	Current Page Pointer
PrevPageLink	Preview Page Pointer with same MOID
NextPageLink	Next Page Pointer with same MOID
MOID	Moving Object Identifier
DataLength	The number of locations stored in it
MBR	Min Bounding Rectangle of Locations in Page
FromTime	time that first location in page
ToTime	time that last location in page
MOTLocations	*DataLength*-MOTLocations

5. THE IMPLEMENTAION

5.1 Connect Manager

When client could connect File-based Storage System with IP and Port, Connection Manager returns Simple Query Processor. Connection Manager is module which providers connection information of that Simple Query Processor. It has function to connect with client by opening channel, to active SQP processing client's request.

5.2 5.2 Simple Query Processor

Simple Query Processor has a function to access to client, to analyze the query requested by client, to process analyzing query using Buffer Manager and Data Manager. Main function of Simple Query Processor is like table X. therefore, it do create table, delete table, insert MORow, search locations with multifarious option.

```
public string CreateTable(string tableName)
public string DeleteTable(string tableName)
public bool InsertRow(MORow moRow)
public void Delete(MOID id)
public IEnumerator IDSearch()
public IEnumerator IDSearch(MOID[] ids)
public IEnumerator IDSearch(MOID[] ids, MOPeriod period)
public IEnumerator Search_PrevMOTLocation(MOID id,
                        MOTime Time, int num)
public IEnumerator Search_NextMOTLocation(MOID id,
                        MOTime Time, int num)
```

Figure 8. Function of Simple Query Processor

5.3 Buffer Manager

Buffer Manager which is composed of a set of MORow objects is using hash-table. The key of hash-table is MOID. And the number of Buffer Manager is equal to the number of table. Each Buffer Manager is connected to File Manager (Log, Meta, Location, Index) like Figure 5. It manages recently Location Information for time interval by MOID.

5.4 Location Manager

Location Manager is module to store MORow that is set of locations into physical file permanently by MOPage. And it do retrieve physical file by MOPage. Data file is composing of Head Block and Body Block like Table 2. Head Block is composed like Table2. Head Block is information about location data file. And In Figure 9, we show the default function of Log Manager. For example, the function of LocalCreateDataFile create data file, the function of write store MOPage. And besides it support the function to retrieve by the various conditions.

Table 2. Data file Structure

Head Block	
HeaderSize	The size of HeadBlock
Version	The version of file
FileID	Data File Identifier
PageLength	The number of Page
MaxDataLength	the max number of locations stored in a Page
PageSize	The size of Page(by byte)
LastPageNumber	Page Pointer that is updated lastly
LastDeletePage	Page Pointer that is deleted
MBR	Min Bounding Rectangle of Locations in file
FromTime	Time that first location in file
ToTime	time that last location in file
Body Block	
MOPages	Set of MOPage

```
private int LocalCreateDataFile()
public  int Write(MOPage page)
private bool LocalUpdateIndexFile(MOPage page)
private MOPage LocalReadPage(PageLink pLink)
public MORow SearchAll(MOID moID)
public MOTLocation SearchFirst(MOID moID)
public MORow Search(MOID moID, MOTime fromTime,
                MOTime toTime)
public MORow PreviousSearch(MOID moID, MOTime
                fromTime, uint length)
public MORow NextSearch(MOID moID, MOTime fromTime,
                uint length)
public int Delete(MOID moID)
```

Figure 9. Function of Location Manager

5.5 Index Manager

Index Manager plays a role to create index file. In Figure 4. Indexing is about Moving Object. But it manages index-information of MOPage. Index data file exist per table. Index key is MOID, MOTIME. And corresponding data of Key is a set of PageID that is offset of Index file.

```
public bool AddIndex(MOID moID, PageLink firstPage)
public bool UpdateIndex(MOID moID, PageLink lastPage)
public bool DeleteIndex(MOID moID)
public PageLink SearchFirstPage(MOID moID)
public PageLink SearchLastPage(MOID moID)
public PageLink SearchPage(MOID moID,MOTime moTIME)
public PageLink SearchPage(MBR mbr)
```

Figure 10. Function of Index Manager

5.6 Log Manager

Log Manager create log file and log buffer. The size and contents of buffer are made a decide with mata-data of database. A type of log content is three kind that is no message, simple message, and detail message. First, it insert message to log buffer. Second, if log buffer is full, log manager write log buffer to file.

```
private int LocalWrite(int logBufferNumber)
public int AddLog(string query, bool success, MORow morow,
            string message)
public int ChangeLogBuffer()
public string ChangeLogFile()
```

Figure 11. Function of Log Manager

5.7 Meta-data Manager

To work this system, it needs meta-data. So, meta-data manager is manaing their meta-data. Each database has meta-data of database and meta-data of Tables. Each structure is like Table3 and Table 4.

Table 3. Database Meta-data

DBName	Name of Database
DBPath	Path of Database
DBProcessorLength	The number of Processor
DBProcessorPorts	List of Port
DBPorts	The number of Port
LogFileName	FileName for Log
LogOption	Option of Log
LogBufferLength	The number of Log Buffer
StorageMBR	MBR of DB
MOTLength	Total number of MOTLocation
Shutdown	Flag for shotdown if it is OK

Table 4. Table Mata-data

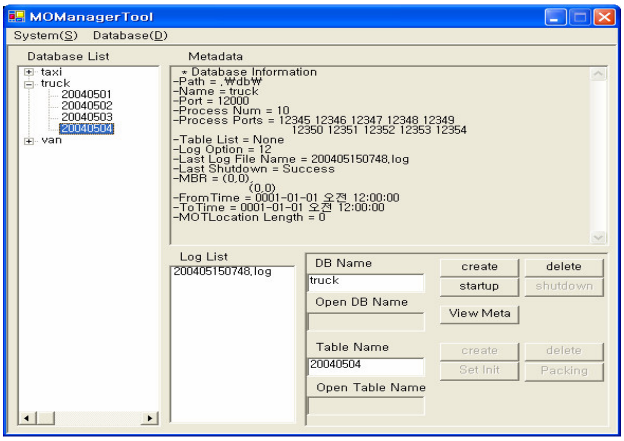
TableName	Table Name
FileID	Data File Identifier
FileName	Data File Name
MBR	Min Bounding Rectangle of Locations in Table
FomeTime	time that first location in Table
ToTime	time that last location in Table
PageSize	Page Size of data file

```
public int WriteDBMetadata()
private int LocalReadDBMetadata()
public long Add(string tableName, long fileID, string
        fileName, int pageSize)
public int Update(string tableName, long fileID, MOMBR
        mbr, MOTime from, MOTime to)
```

Figure 11. Function of Meta Manager

5.8 Admin Tool

Admin Tool is operating and managing on File-Based Storage System. For example, it is to do create, delete, start and stop database. In Addition, it is to create table and to delete table. It is retrieve database list, table list in each database, log list, log contents, metadata information. And it is to make a conversion between this system and tradition database (ORACLE, DB/2, SQLServer etc). Interface of Admin-Tool is like Figure 12.



```
 * Database Information
-Path = .₩db₩
-Name = truck
-Port = 12000
-Process Num = 10
-Process Ports = 12345 12346 12347 12348 12349
                 12350 12351 12352 12353 12354
-Table List = None
-Log Option = 12
-Last Log File Name = 200405150748.log
-Last Shutdown = Success
-MBR = (0,0),
        (0,0)
-FromTime = 0001-01-01 오전 12:00:00
-ToTime = 0001-01-01 오전 12:00:00
-MOTLocation Length = 0
```

Log List: 200405150748.log

DB Name: truck

Open DB Name:

Table Name: 20040504

Open Table Name:

Figure 12. Admin Tool

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

Erwig, M., Guiting, R. H., Schneider, M., and Vazirgiannis, M., 1999. "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Obejct in Databases," GeoIn-fomatica, Vol.3, No.3, pp.269-296.

Forlizzi, L., Güting, R. H., Nardelli, E., and Schneider, M., 2000. "A Data Model and Data Structures for Moving Objects Databases," Proc. ACM SIGMOD Conf. (Dallas, Texas), pp. 319-330.

Güting, R. H., Böhlen, M. H., Erwig, M., Jensen, C. S., Lorentzos, N. A., Schneider, M., and Vazirgiannis, M., 1998. "A Foundation for Representing and Querying Moving Objects," Fern Universität Hagen, Informatik-Report 238, September 1998, ACM Transactions on Database Systems, 25(1), pp. 1-42.

Guttman, A., 1984. "R-trees: A dynamic index structure for spatial searching," ACM SIGMOD Conference, pp. 47-54.

Jensen, C. S., Friis-Christensen, A., Pedersen, T. B., Pfoser, D., Saltenis, S., and Tryfona, N., 2001. "Location-Based Services - A Database Perspective,"' Proceedings of the Eighth Scandinavian Research Conference on Geographical Information Science, As, Norway, June 25-27, pp. 59-68.

Kollios, G., Gunopulos, D., and Tsotras, V. J., 1999. "On Indexing Mobile Objects," ACM Symposium on Principles of Database Systems, pp. 261-272.

Nascimento M. A., and Silva, J. R. O., 1998. "Towards historical R-trees," ACM SAC.

Pfoser, D., and Theodoridis, Y., 2000. "Generating Semantics-Based Trajectories of Moving Objects," International Workshop on Emerging Technologies for Geo-Based Applications, Ascona, Switzerland.

Pfoser, D., Jensen, C. S., and Theodoridis, Y., 2000. "Novel Approaches in Query Processing for Moving Object Trajectories," VLDB 2000, pp. 395-406.

Song, Z., and Roussopoulos, N., 2001. "Hashing Moving Objects," MDM 2001, LNCS 1987, pp. 161-17.

Wolfson, O., Xu, B., Chamberlain, S., and Jiang, L., 1998. "Moving Objects Databases: Issues and Solutions," SSDBM 1998, pp. 111-122.

Wolfson, O., Chamberlain, B. X. S., Sistla, P., Xu, B., and Zhou, X., 1999. "DOMINO: Databases fOr MovINg Objects tracking," ACM International Conference on SIGMOD , pp. 547-549.

SoftBank Research, IT Insight Strategy Report,"LBS, Now & Future"