Journal of University of Science and Technology Beijing
Volume 9, Number 2, April 2002, Page 156

An architecture for mobile database management system

Dong Li and Yucai Feng
Computer School, Huazhong University of Science and Technology, Wuhan 430074, China
(Received 2001-01-20)

Abstract: In order to design a new kind of mobile database management system (DBMS) more suitable for mobile computing than the existent DBMS, the essence of database systems in mobile computing is analyzed. An opinion is introduced that the mobile database is a kind of dynamic distributed database, and the concept of virtual servers to translate the clients' mobility to the servers' mobility is proposed. Based on these opinions, a kind of architecture of mobile DBMS, which is of versatility, is presented. The architecture is composed of a virtual server and a local DBMS, the virtual server is the kernel of the architecture and its functions are described. Eventually, the server kernel of a mobile DBMS prototype is illustrated.

Key words: mobile database, dynamic distributed database, DBMS, architecture, virtual server, data region

The rapidly development of the technologies of cellular communications, wireless LAN, and satellite services promises to make it possible for mobile users to access information anywhere and at any time. Mobile computers, such as laptop computers and notebooks can be connected to a fixed information network to retrieve and process information. The resulting computing environment is often called mobile or nomadic computing, no longer requires a user to maintain a fixed position in the network and enables a user to maintain a unfixed position in the network and enable almost unrestricted user mobility.

There are already a number of mobile applications of mobile wireless computing. These applications promote advances of the database technology. Mobile computing, WEB computing, data warehouse and data mining are becoming to be three technical ways by which an application of "client as a center" is completed. Recently, portable computers such as palmtop computers, notebook computers, PDAS *etc.*, have been developed and applied popularly, accompanying with development of the mobile communication technology, which lead to the emergence of mobile computing environments. But mobile computing proposes new issues to network communication and database management. A mobile computing environment is an extension of the traditional distributed computing environment. Its typical architecture [1] includes three main components: MU (Mobile Unit for short, such as PDA, *etc.*), MSS (Mobile Service Support, a fixed point supporting mobile computing with wireless communication interface) and FH (Fixed Host, a fixed host without wireless communication interface).

A mobile computing environment has remarkable characteristics, of those the mobility and disconnection are the most two important characteristics [2,3]. Firstly, in the traditional Distributed Database Management System (DDBMS for short), the scheduling and service allotment of system resources are based on the comparatively static and stable environment. While in a changing environment, the system would be unstable. Secondly, a mobile database is usually location-dependent and time-dependent, but the traditional DDBMS can't satisfy the requirement. Thirdly, a mobile computer connects and disconnects frequently and predictably, but the traditional DBMS can't provide this kind of services. Therefore, it is needed to work out a new kind of mobile database architecture that can describe and define the location-dependent data caused by the motion of clients, and design a new kind of DBMS suitable to manage the database in a mobile computing environment.

1 Mobile database architecture

In a mobile computing environment, the movement of a client implies to access different database servers. If assuming the client static, it can be said that these database servers are "moving" due to the relativity of mobility. Here we introduce the concept of a Virtual Server (VS for short) shown in **figure 1**.

The same MU can be serviced by S_1 and S_2 in the site *A*, while in the site *B* serviced by S_2 and S_3, although S_1, S_2 and S_3 are inter-connected physically. In

Corresponding author: Dong Li **E-mail:** lidong@public.wh.hb.cn

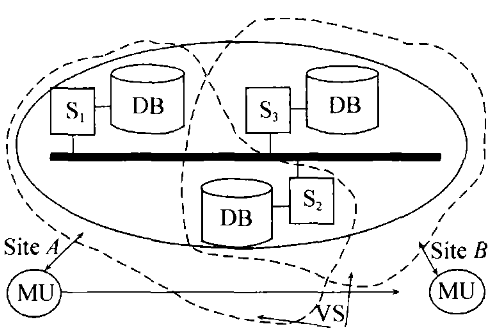


Figure 1 Virtual server. S_1, S_2, S_3 — Site server; MU — Mobile unit; DB — database; VS — virtual server.

different sites, the same query issued by the MU may-be gets different but correct values, and the MU gets different services from different DB Servers. It can be said that the MU makes a 'virtual' connecting with the different DB servers, *i.e.*, there is a "virtual server" for the MU. One virtual server services one "data region" (the concept will be introduced in the following). One data region only has one correct value to the client. Due to movement of the client, the querying results are different but correct within different data regions. It implies that the database is location-dependent. From the figure 1, we can see that data is not only distributed but also changing dynamically. So we introduce the concept "dynamic distributed database". The dynamic distributed database is that the static distributed database in physical, but for the client, it changes dynamically in logic along with the client's moving. With this kind of opinion, it is thought that the DBMS in a mobile computing environment is a dynamic distributed DBMS called the mobile DBMS. In order to describe the location dependency of a mobile database, we introduce the concept "data region schema" shown as **figure 2**.

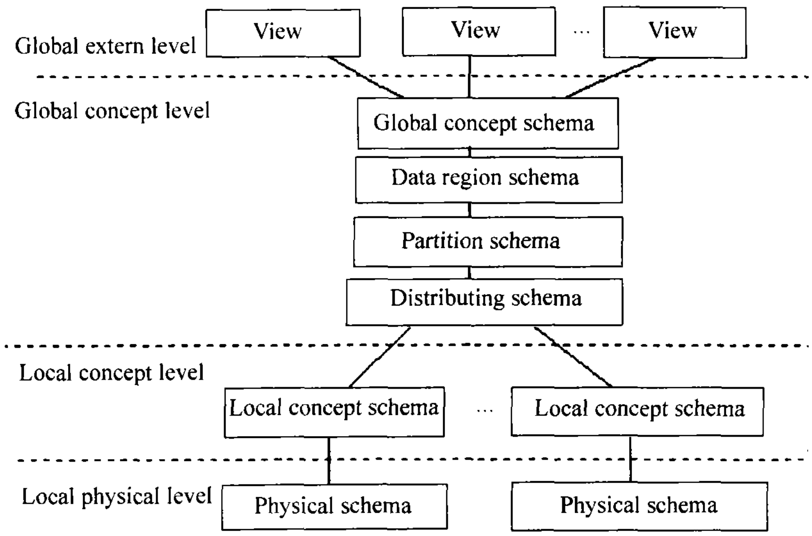


Figure 2 A mobile database architecture.

In the mobile database, the data region schema describes data region partition information, such as region number, region id, the name of database schema in the data region, relation of different data regions in the mobile database and so on. The partition and distributing schema are the same as those of the traditional distributed database system.

In order to discuss the data region schema and the translation concerned, we introduce several definitions first.

Definition 1. The physical region G is the entire area covered by the mobile computing platform, thus a mobile unit can freely move around in G.

Definition 2. A location L is a precise point within the physical region. It stands for the smallest identifiable position in G. Each location is identified by a specific id, L and $G = \bigcup L$.

Shown as figures 1 and 2, the MU moves from L_A (at A) to L_B (at B) in the same physical region, but in the different data region. It is due to the data region that the same query maybe gets different but both correct values at different locations, L_A and L_B. In order to describe the concept of data region, we should firstly introduce two kinds of replications.

Definition 3. The temporal replication refers to copies of data objects all of which have only one consistent data value at any point in time. One of these copies is called a temporal replica.

Definition 4. The spatial replication refers to copies of data objects which may have different correct data value at any point in time. Each value is correct within a given location area. One of these copies is called a spatial replica.

In the mobile database, the replicas at a cache of the MU are really temporary replicas. It implies that there exist temporary replicas of spatial replicas.

The mobility of an MU only affects the spatial replication. The spatial replication introduces several new issues: the definition of location, data consistency, and how to specify location dependent query. Till now we can give the definition of data region.

Definition 5. A data region *R* is an area within *G*, satisfying $R \subseteq G$. In the data region *R*, one correct value exists for a spatial replica.

Figure 1 illustrates this opinion. Generally, the temporal replication is the same as the traditional distributed replica, and the spatial replication is location dependent replica. In one data region, there are no replication occurs. The distributed replicas are copies of each other, which may have different values temporarily but there is only one correct value. The location-dependent data have multiple copies and multiple correct values. The correct value is determined by location. One of most differences is that there exists a spatial replication in the mobile database, but no spatial database in the traditional distributed database. The difference between the spatial database and the mobile database is discussed in reference [4]. In order to decide the relation among data objects, data region and location, the translation between them can be described by Definitions 6 and 7.

Definition 6. Given a set of data objects *D* and a set of data region *R*, a data location mapping DLM is a mapping DLM: $D \to R$, where DLM(D) = $\{R_1, R_2, \cdots, R_n\}$, $R_i \in R$,

$$\bigcup_{i=1}^{n} R_i = G,$$

and $\forall i, j, R_i \cap R_j = \Phi$.

It implies that the set of data region for data objects is a partitioning of the physical domain.

Definition 7. Given a set of data objects *D*, a set of location ids *L*, and a set of data regions *R*, a data region mapping DRM is a mapping DRM: $D \times L \to R$ where DRM ($\langle D, L \rangle$) $\in$ DLM(D).

It can be seen that a data region is identified by the data object and location. So as an MU moves, each location uniquely identifies for all data objects the data region which each belongs to. This is the basis to perform location dependent queries. Normally, a location dependent query returns data values for the location where it executes.

By the data region schema, we can describe the location-dependent data in the mobile database.

In addition, there is a translation between the data region schema and partition schema, which translates the global database into several region databases.

2 Mobile database management system prototype

Based on the architecture and the concept of virtual server, we construct a mobile DBMS architecture in the mobile computing environment shown as **figure 3**.

LDBMS provides the basic DBMS functions, which is similar to those of the traditional DBMS. VS stands for virtual server and realizes the translation between the data region schema. and the partition schema. Its main functions are:

(1) Offering the client interface and communication interface.

(2) Constructing the dynamic distributed database and database locating function. The dynamic distributed database can describe the location-dependent information, and has the ability of summary, extracting and abstracting on original data. The database locating function is able to find the logic location of the physical database.

(3) Choosing processing strategy. Specify a task whether will be processed at a local place or at a remote place, whether processing in one node or collaborating with other nodes.

(4) Translation processing.

(5) Mobile transaction processing.

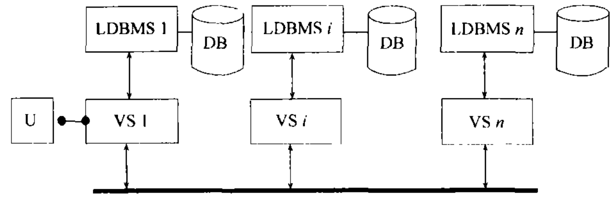
(6) Mobile query and optimization.

(7) Replica and cache management.

Regardless of the constructing of the dynamic distributed database in (2) and the functions of (5) and (6), the VS would be similar to the GDBMS of the traditional DDBMS, and the mobile DBMS would act as a traditional DBMS. Furthermore, if getting rid of (5) and (6) and simplifying the functions in (1)-(3), the VS would act as a WebServer, and the application system based on the mobile DBMS would be an application to Browse/Server. From figure 3 it can be seen that the architecture is based on the Share Nothing (SN for short) architecture, therefore this architecture implies to have a good scalability. According to the architecture, a mobile database management prototype MDM2 is designed based on DM2.

DM2 [5] is a DDBMS with C/S architecture developed by Huazhong University of Science and Technology in 1996. It adopts a relation-hierarchy model and conforms to SQL92. Besides the basic SQL functions, DM2 also supports some multimedia data types and GIS functions. It provides with ODBC and JDBC interface, so we can use current popular development tools such as Power Builder to develop the database application system quickly. Its security level is B1 and till now we have developed several MIS (Management Information System, MIS) applications based on DM2 successfully. The architecture for MDM2 server is shown as **figure 4**.

The server of MDM2 can be divided into two parts: the DM2 server kernel and the virtual server. The first part provides basic functions such as accessing of multimedia data, GIS data, I/O process scheduling and so on. The second part, virtual server, also contains several components: router management module, communication module, translation module, loading balancing module, mobile transaction management module, mobile query module, data buffering management module, *etc.* In this paper we only introduce its several key technologies.



LDBMS *i* —local DBMS in the site *i*; VS *i* —virtual server in the site *i*; U — mobile client;

Figure 3 A mobile database management system.

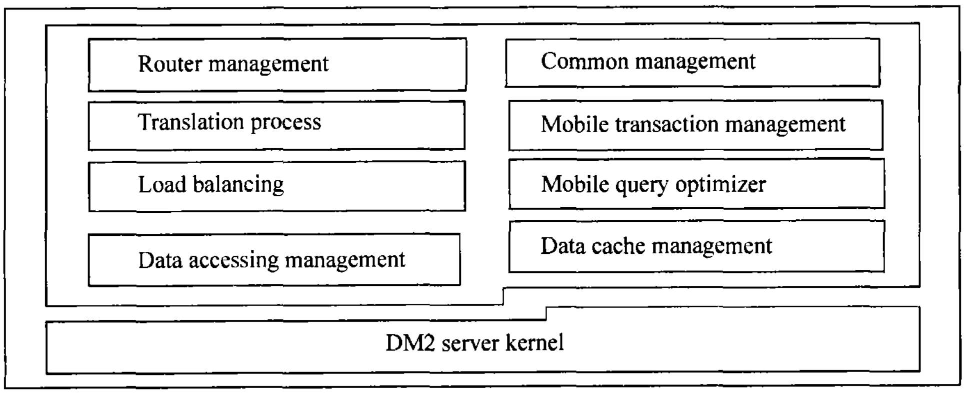


Figure 4 Architecture of MDM2 server.

Mobile transaction processing. The mobile transaction processing should solve the transaction processing under the condition of frequent and foreseeable disconnection in the mobile computing environment. The traditional strategy and algorithm for transaction management are not suitable to the mobile transaction processing completely. In order to ensure the successful committing of transactions, we must design and implement new kinds of strategies and algorithms for transaction processing. From these opinions, the mobile transaction processing can adopt the following principles, which is the modified version of previous works [1]:

(a) Confirming whether migrating transactions or not, according to the threshold of a operating time interval, *i.e.*, whether to migrate the transaction from the mobile computer onto the fixed computer to execute or not.

(b) Confirming whether to download the data to execute or not according to the threshold of data size (can be set up at first), *i.e.*, whether to download and execute in the mobile computer or in the fixed computer.

(c) Backing up the log file in the fixed computer and maintaining the consistence between the mobile computer and the fixed computer.

(d) Announcing its status actively when the mobile computer is about to disconnect.

Mobile query processing. Mobile database querying results always depend on its location when the client is moving. For example there is a query Q_1 = "query all gas station around the client within 5 km". Of course, the results of Q_1 are likely to vary from one place to another place. Due to the client's motion, the query language of the system should support two kinds of linguistic such as MAY and MUST [6]. For example, there is another query Q_2 = "query all available hotel around the location where the client will reach with 0.5 km in 10 min later". Because the client may make a moving not only according to a existing motion plan but also adopting a new motion plan, the query is only a possibility query and the results may be true or not. This kind of query can be called MAY query. Of course, there is another kind of query called MUST query. So the mobile DBMS should support these two kinds of query linguistic and provide abilities of spatial and temporal query.

Due to the moving of mobile clients, the query and updating on the mobile data become very complicated. One difficulty of those is to locate the positions of mobile clients. If inefficient methods adopted, it may cause heavy overload because of the large number of moving clients. In order to locate a mobile object efficiently, the mobile DBMS would be better to have the ability of GIS to translate physical location of the mobile object into logic location in database. DM2 has an embedded GIS function, which is the research base of our mobile DBMS project. In order to execute the query easily, we develop a GUI tool for querying the information around mobile clients. You can commit a query by selecting a region around a client, then system executes the operation and return the results to the client.

Data buffering management. The database duplication technology can improves reliability and usability of the system, and is one way to solve mobility and disconnection in the mobile computing environment. We consider providing the cache between the mobile computers and virtual servers to solve the mobility and disconnection. An incremental updating method is adopted to maintain consistency of the database. In the cache of a mobile client, there is a database mobile subset and an incremental value subset. When in the status of online, the client accesses the server periodically to get the newest database version to keep consistency. When updating the database, the mobile client first generates the incremental value locally and puts it into the incremental subset queue. Then the client packets the incremental values and logs, and sends to the server. Once accepting the data packages, the server decomposes and puts them into task queue and data queue. The updating of the database is under the control of server kernel. Sequentially, it is up to the server to schedule and handle. If the mobile client is in the status of offline, the incremental updating of the database is buffered in a local database of the client. Once connected, then the mobile client packets the updating value subset and sends them to the server. This method can decrease data communication and improve system transaction throughput and parallel ability. For the database cache consistency maintenance, we adopt a kind of asynchronously avoidance-based algorithm.

3 Conclusion

The traditional distributed DBMS can't work efficiently in a mobile computing environment. Therefore, it is necessary to design a new kind of DBMS to support data management in the mobile computing environment. This paper introduces the concepts of "virtual server" and "dynamic distributed database", proposes an architecture for mobile database based on these, and presents a system prototype MDM2 based on the DM2.

References

[1] M.H. Dunham and A. Helal, Mobile computing and database: anything news [J], *SIGMOD Record*, 24(1995), No.4, p.5.

[2] R. Alonso and H.F. Korth, Database system issue in nomadic computing, [in] *Proceeding of the 1993 ACM SIGMOD International Conference on Management of Data* [C], Washington, DC, 1993, p.388-392.

[3] T. Imielinski and B.R. Basrinath, Mobile wireless computing [J], *Communication of ACM*, 37(1994), No.10, p.19.

[4] O. Wolfson, B. Xu, S. Chamberlain, *et al.*, Moving objects databases: issues and solutions, [in] *Proceedings of the 10th International Conference on Scientific and Statistical Database Management* [C], Capri, 1998, p.111-122

[5] Y.Y. Wu, Y.C. Feng, and F.M. Yang, Research and development of a distributed multimedia database management system [J], *Computer Research and Development* (in Chinese), 34(1997), No.1, p.28.

[6] A.P. Sistla, O. Wolfson, and S. Chamberlain, Modeling and querying moving objects, [in] *Proceedings of the 13th International Conference on Data Engineering* [C], Birmingham, 1997, p.422-432.