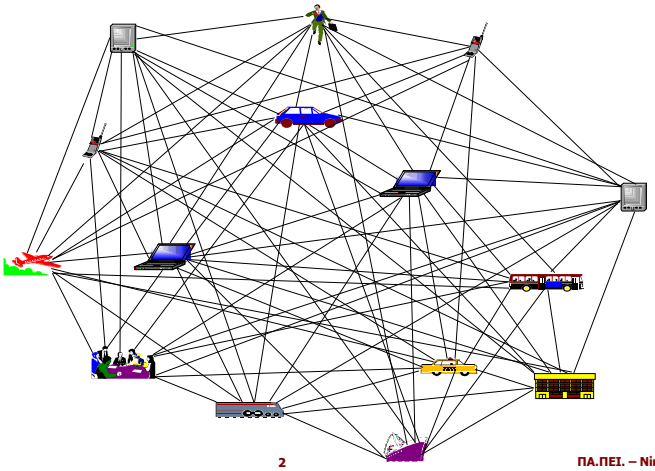
Κινητές ΒΔ

Mobile Database Systems

Βασική πηγή διαφανειών: Vijay Kumar, Mobile Database Systems, J.Wiley & Sons, 2006.
Εργαστήριο Πληροφοριακών Συστημάτων, Παν/μιο Πειραιώς (http://infolab.cs.unipi.gr/)
έκδοση: Ιανουάριος 2010

Mobile Database Systems

Fully connected information space



Fully connected information space

Can be created and maintained by integrating legacy database systems, and wired and wireless systems (PCS, Cellular system, and GSM)

What is a Mobile Database System (MDS)?

A system with the following structural and functional properties

- Distributed system with *mobile connectivity*
- Full database system capability
- Complete spatial mobility
- Built on PCS/GSM platform
- Wireless and wired communication capability

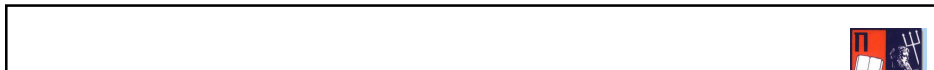
What is a mobile connectivity?

A mode in which a client or a server can establish communication with each other whenever needed. ***Intermittent connectivity*** is a special case of mobile connectivity.

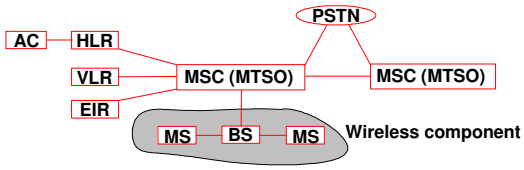
What is intermittent connectivity?

A node in which only the client can establish communication whenever needed with the server but the server cannot do so.

Personal Communication System (PCS)



A system where wired and wireless networks are integrated for establishing communication.

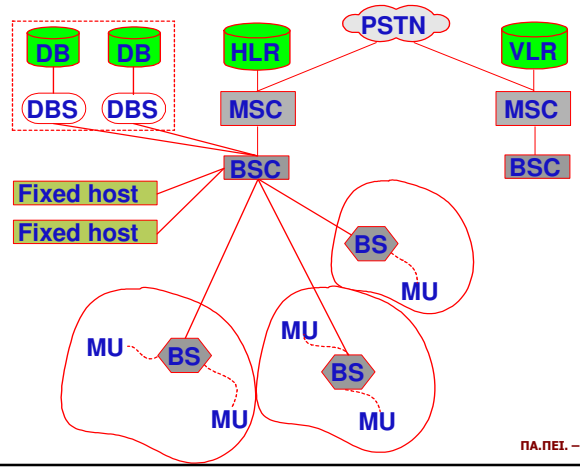


PSTN: Public Switched Network.
MSC: Mobile Switching Center. Also called MTSO (Mobile Telephone Switching Office).
BS: Base Station.
MS: Mobile Station. Also called MU (Mobile Unit) or Mobile Host (MH).
HLR: Home Location Register.
VLR: Visitor Location Register.
EIR: Equipment Identify Register.
AC: Access Chanel.

Mobile Database Systems (MDS)

- Architecture
- Data categorization
- Data management
- Transaction management
- Recovery

A Reference Architecture (Client-Server model)



MDS Applications

- Insurance companies
- Emergencies services (Police, medical, etc.)
- Traffic control
- Taxi dispatch
- E-commerce
- Etc.

MDS Limitations

- Limited wireless bandwidth
- Wireless communication speed
- Limited energy source (battery power)
- Less secured
- Vulnerable to physical activities
- Hard to make theft proof.

MDS capabilities

- Can physically move around without affecting data availability
- Can reach to the place data is stored
- Can process special types of data efficiently
- Not subjected to connection restrictions
- Very high reachability
- Highly portable

Objective

To build a truly ubiquitous information processing system by overcoming the inherent limitations of wireless architecture.

MDS Issues

- Data Management
  - Data Caching
  - Data Broadcast (Broadcast disk)
  - Data Classification
- Transaction Management
  - Query processing
  - Transaction processing
  - Concurrency control
  - Database recovery

MDS Data Management Issues

How to improve data availability to user queries using limited bandwidth?

Possible schemes

- Semantic data caching: The cache contents is decided by the results of earlier transactions or by *semantic data set.*
- Data Broadcast on wireless channels

MDS Data Management Issues

How to improve data availability to user queries using limited bandwidth?

Semantic caching

- Client maintains a semantic description of the data in its cache instead of maintaining a list of pages or tuples.
- The server processes simple predicates on the database and the results are cached at the client.

MDS Data Management Issues

Data Broadcast (Broadcast disk)

A set of most frequently accessed data is made available by continuously broadcasting it on some fixed radio frequency. Mobile Units can tune to this frequency and download the desired data from the broadcast to their local cache.

A broadcast (file on the air) is similar to a disk file but located on the air.

MDS Data Management Issues

Data Broadcast (Broadcast disk)

The contents of the broadcast reflects the data demands of mobile units. This can be achieved through data access history, which can be fed to the data broadcasting system.

For efficient access the broadcast file use index or some other method.

MDS Data Management Issues

How MDS looks at the database data?

Data classification

- **Location Dependent Data (LDD)**
- **Location Independent Data (LID)**

MDS Data Management Issues

Location Dependent Data (LDD)

The class of data whose value is functionally dependent on location. Thus, the value of the location determines the correct value of the data.

Location ⟶ Data value

Examples: City tax, City area, etc.

MDS Data Management Issues

Location Independent Data (LID)

The class of data whose value is functionally independent of location. Thus, the value of the location does not determine the value of the data.

Example: Person name, account number, etc. The person name remains the same irrespective of place the person is residing at the time of enquiry.

MDS Data Management Issues

Location Dependent Data (LDD)

Example: Hotel Taj has many branches in India. However, the room rent of this hotel will depend upon the place it is located. Any change in the room rate of one branch would not affect any other branch.

Schema: It remains the same only multiple correct values exists in the database.

MDS Data Management Issues

Location Dependent Data (LDD)

LDD must be processed under the location constraints. Thus, the tax data of Pune can be processed correctly only under Pune's finance rule.

Needs *location binding* or *location mapping* function.

MDS Data Management Issues

Location Dependent Data (LDD)

Location binding or *location mapping* can be achieved through database schema or through a location mapping table.

MDS Data Management Issues

Location Dependent Data (LDD) Distribution

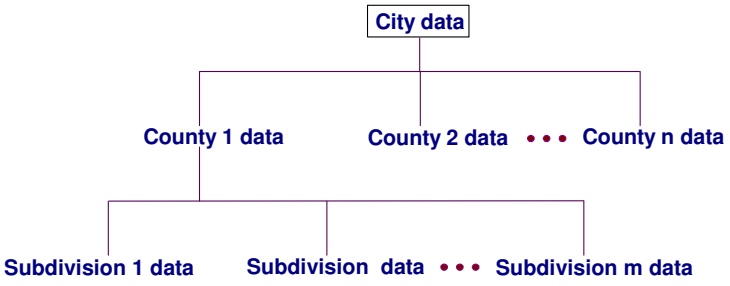
MDS could be a federated or a multidatabase system. The database distribution (replication, partition, etc.) must take into consideration LDD.

One approach is to represent a city in terms of a number of mobile cells, which is referred to as "Data region". Thus, Pune can be represented in terms of *N* cells and the LDD of Pune can be replicated at these individual cells.

MDS Data Management Issues

Concept Hierarchy in LDD

In a data region the entire LDD of that location can be represented in a hierarchical fashion.



MDS Query processing

Query types

- Location dependent query
- Location aware query
- Location independent query

MDS Query processing

Location dependent query

A query whose result depends on the geographical location of the origin of the query.

Example

What is the distance of Pune railway station from here?

The result of this query is correct only for “here”.

MDS Query processing

Location dependent query

Situation: Person traveling in the car desires to know his progress and continuously asks the same question. However, every time the answer is different but correct.

Requirements: Continuous monitoring of the longitude and latitude of the origin of the query. GPS can do this.

MDS Transaction Management

Transaction properties: **ACID** (Atomicity, Consistency, Isolation, and Durability).

Too rigid for MDS. Flexibility can be introduced using workflow concept. Thus, a part of the transaction can be executed and committed independent to its other parts.

MDS Transaction Management

Transaction fragments for distributed execution

Execution scenario: User issues transactions from his/her MU and the final results comes back to the same MU. The user transaction may not be completely executed at the MU so it is fragmented and distributed among database servers for execution. This creates a Distributed mobile execution.

MDS Transaction Management

A mobile transaction (MT) can be defined as

T_i is a triple $<F, L, FLM>$; where

$F = \{e_1, e_2, \ldots, e_n\}$ is a set of execution fragments,

$L = \{l_1, l_2, \ldots, l_n\}$ is a set of locations, and

$FLM = \{flm_1, flm_2, \ldots, flm_n\}$ is a set of fragment location mapping where $\forall j, flm_i (e_i) = l_i$

MDS Transaction Management

An execution fragment e_{ij} is a partial order $e_{ij} = \{\sigma_i, \preceq_i\}$ where

- $\sigma_i = OS_j \cup \{N_i\}$ where $OS_j = \cup_k O_{jk}$, $O_{jk} \in \{read, write\}$, and $N_j \{Abort_L, Commit_L\}$.
- For any O_{jk} and O_{jl} where $O_{jk} = R(x)$ and $O_{jl} = W(x)$ for data object x, then either $O_{jk} \preceq_j O_{jl}$ or $O_{jl} \preceq_j O_{jk}$.

MDS Transaction Management

Mobile Transaction Models

Kangaroo Transaction: It is requested at a MU but processed at DBMS on the fixed network. The management of the transaction moves with MU. Each transaction is divided into subtransactions. Two types of processing modes are allowed, one ensuring overall atomicity by requiring compensating transactions at the subtransaction level.

MDS Transaction Management

Mobile Transaction Models

Reporting and Co-Transactions: The parent transaction (workflow) is represented in terms of reporting and co-transactions which can execute anywhere. A reporting transaction can share its partial results with the parent transaction anytime and can commit independently. A co-transaction is a special class of reporting transaction, which can be forced to wait by other transaction.

MDS Transaction Management

Mobile Transaction Models

Clustering: A mobile transaction is decomposed into a set of weak and strict transactions. The decomposition is done based on the consistency requirement. The read and write operations are also classified as weak and strict.

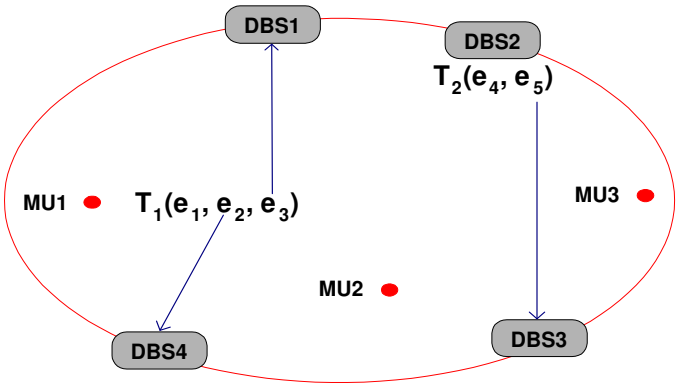
MDS Transaction Management

Mobile Transaction Models

Semantics Based: The model assumes a mobile transaction to be a long lived task and splits large and complex objects into smaller manageable fragments. These fragments are put together again by the merge operation at the server. If the fragments can be recombined in any order then the objects are termed *reorderable* objects.

MDS Transaction Management

Mobile Transaction execution.



MDS Transaction Management

Serialization of concurrent execution.

- Two-phase locking based (commonly used)
- Timestamping
- Optimistic

Reasons these methods may not work satisfactorily

- Wired and wireless message overhead.
- Hard to efficiently support disconnected operations.
- Hard to manage locking and unlocking operations.

MDS Transaction Management

Serialization of concurrent execution.

New schemes based on timeout, multiversion, etc., may work. A scheme, which uses minimum number of messages, especially wireless messages is required.

MDS Transaction Management

Database update to maintain global consistency.

Database update problem arises when mobile units are also allowed to modify the database. To maintain global consistency an efficient database update scheme is necessary.

MDS Transaction Management

Transaction commit.

In MDS a transaction may be fragmented and may run at more than one nodes (MU and DBSs). An efficient commit protocol is necessary. 2-phase commit (2PC) or 3-phase commit (3PC) is no good because of their generous messaging requirement. A scheme which uses very few messages, especially wireless, is desirable.

MDS Transaction Management

Transaction commit.

One possible scheme is "timeout" based protocol.

Concept: MU and DBSs guarantee to complete the execution of their fragments of a mobile transaction within their predefined timeouts. Thus, during processing no communication is required. At the end of timeout, each node commit their fragment independently.

MDS Transaction Management

Transaction commit.

Protocol: **TCOT-Transaction Commit On Timeout**

Requirements

Coordinator: Coordinates transaction commit

Home MU: Mobile Transaction (MT) originates here

Commit set: Nodes that process MT (MU + DBSs)

Timeout: Time period for executing a fragment

MDS Transaction Management

Protocol: TCOT-Transaction Commit On Timeout

- MT arrives at Home MU.
- MU extract its fragment, estimates timeout, and send rest of MT to the coordinator.
- Coordinator further fragments the MT and distributes them to members of commit set.
- MU processes and commits its fragment and sends the updates to the coordinator for DBS.
- DBSs process their fragments and inform the coordinator.
- Coordinators commits or aborts MT.

MDS Transaction Management

Transaction and database recovery.

Complex for the following reasons

- Some of the processing nodes are mobile
- Less resilient to physical use/abuse
- Limited wireless channels
- Limited power supply
- Disconnected processing capability

MDS Transaction Management

Transaction and database recovery.

Desirable recovery features

- **Independent recovery capability**
- **Efficient logging and checkpointing facility**
- **Log duplication facility**

MDS Transaction Management

Transaction and database recovery.

- **Independent recovery capability reduces communication overhead. Thus, MUs can recover without any help from DBS**
- **Efficient logging and checkpointing facility conserve battery power**
- **Log duplication facility improves reliability of recovery scheme**

MDS Transaction Management

Transaction and database recovery.

Possible MU logging approaches

- **Logging at the processing node (e.g., MU)**
- **Logging at a centralized location (e.g., at a designated DBS)**
- **Logging at the place of registration (e.g., BS)**
- **Saving log on Zip drive or floppies.**

Mobile Agent Technology

A mobile agent is an independent software module capable of

- **Migrating to any node on the network**
- **Capable of spawning and eliminating itself**
- **Capable of recording its own history**

Mobile Agent Technology

A mobile agent can be used for the following activities, which are essential for recovery.

- Centralized and distributed logging
- Log carrier. A Mobile unit may need to carry its log with it for independent recovery
- Log processing for database recovery
- Transaction commit or abort

Mobile E-commerce

What is E-commerce?

Mapping of business activity on the network. The network may be mobile of ad-hoc units, in which case the scope of business activities significantly increases.

Mobile E-commerce

Why mobile E-commerce?

To make business activity free from spatial constraints. This allows tremendous flexibility to customers as well as to vendors.

Important gain: Making information available at the right time, at the right location, and in a right format.

Mobile E-commerce

Requirements for a mobile E-system

- Security
- Reliability
- Efficient
- Customer trust
- Quality of service

Mobile E-commerce

These requirements are difficulty and complex to achieve

Security

Conventional key approaches needs revision.

Reliability

Hard to provide mainly because of the unreliability and limitations of resources.

Mobile E-commerce

These requirements are difficulty and complex to achieve

Efficient

This capability can be easily improved mainly because of the elimination of spatial constraints.

Customer trust

A time consuming activity. Customer do not easily trust electronic communication and always wants to see a reliable backup service.

Mobile E-commerce

These requirements are difficulty and complex to achieve

Quality of service

Mobility and web provides ample scope for improving the quality of service. An integration of mobility, web, data warehousing and workflow offers tremendous growth potential and a very controlled way of managing business activities.

Conclusions and summary

Wireless network is becoming a commonly used communication platform. It provides a cheaper way to get connected and in some cases this is the only way to reach people. However, it has a number of easy and difficult problems and they must be solved before MDS can be built. This tutorial discussed some of these problems and identified a number of possible approaches.

Conclusions and summary

The emerging trend is to make all service providing disciplines, such as web, E-commerce, workflow systems, etc., fully mobile so that any service can be provided from any place. Customer can surf the information space from any location at any time and do their shopping, make flight reservation, open bank account, attend lectures, and so on. This is what the wireless technology driving us to.

References

1. Acharya, S., Alonso, R., Franklin, M., and Zdonik, S. Broadcast Disks: Data management for Asymmetric Communication Environments. Proc. ACM SIGMOD Conf., San Jose, May, 1995.
2. Alonso, R., and Korth, H. Database Systems Issues in Nomadic Computing. Proc. ACM SIGMOD International Conf. on management of Data, May 1993.

References

3. Barbara, D., and Imielinski, T. Sleepers and Workaholics: Caching Strategies in Mobile Environments. Proc. ACM SIGMOD Conf., Minneapolis, May, 1994.
4. Chrysanthis, P. K., Transaction Processing in Mobile Computing Environment, in IEEE Workshop on Advances in Parallel and Distributed Systems, October 1993.

References

5. Dhawan, C. Mobile Computing. McGraw-Hill, 1997.
6. Dunham, M. H., Helal, A., and Balakrishnan, S., A Mobile Transaction Model That Captures Both the Data and Movement Behavior, ACM/Baltzer Journal on Special Topics in Mobile Networks and Applications, 1997.
7. Forman, H. George and Zahorjan, J. The Challenges of Mobile Computing, IEEE Computers, Vol. 27, No. 4, April 1994.

References

8. Pitoura, E. and Bhargava, B., Maintaining Consistency of Data in Mobile Distributed Environments. Proceedings of 15th International Conference on Distributed Computing Systems., 1995.
9. Pitoura, E. and Bhargava, B., Building Information Systems for Mobile Environments, Proc. 3rd. Int. conf. on Information and Knowledge Management, Washington, DC, No. 1994.

References

10. Vijay Kumar, "Timeout-based Mobile Transaction Commit Protocol", 2000 ADBIS-DASFAA Symposium on Advances in Databases and Information Systems, Prague, Sep. 5-8, 2000.
11. Shaul Dar, Michael Franklin, Bjorn T. Johnsson, Divesh Srivastava, and Michael Tan, "Semantic Data Caching and Replacement", Proc. Of the 22nd VLDB Conference, Mumbai, India, 1996.

References

12. E. Pitoura and G. Samaras, "Data Management for Mobile Computing", Kluwer Academic Publishers, 1998.
13. E. Turban, at. el., "Electronic Commerce: A Managerial Perspective", Prentice Hall, 2000.
14. L. Loeb, "Secure Electronic Transactions", Artech House, 1998.