CACHING ESTRATEGIES FOR MOBILE DATABASES

Heloise Manica
Universidade Federal de Santa Catarina, Florianópolis, Brasil

Murilo S. de Camargo
Universidade de Brasília, Brasília, Brasil.

Keywords: Mobile Computing, Mobile Databases, Data Management, Cache replacement, Cache consistency.

Abstract: Caching remote data in local storage of a mobile client has been considered an effective solution to improve system performance for data management in mobile computing applications. In this paper, we propose a taxonomy for cache management in mobile database systems. The aim is to provide a unifying framework for the problem of caching in mobile computing, then a comparative review of the work done in this area up to now. Such a framework, with the associated analysis of the existing approaches, provides a basis for identifying strengths and weaknesses of individual methodologies, as well as general guidelines for future improvements and extensions.

1 INTRODUCTION

In a mobile computing environment, the clients are mobile units (MUs) that communicate with data servers through a wireless link, accessing information at anytime and anywhere. Several new applications such as traveler information systems, sales in shopping center and train station (Hara, 2002) have been motivating researches on query processing in mobile databases systems (MDB).

The data request is made through equipments as personal digital assistants (PDA), notebooks, etc. that have limited local resources. Moreover, the wireless network is vulnerable to frequent disconnections, low-quality communication and scarce bandwidth.

An effective solution to deal with these problems is the data caching technique that stores data copies frequently used in the clients. The cache management considers two dimensions: consistency and replacement policies. Cache invalidation aims to keep data consistency between the client's cache and the server. The cache replacement policy determines which data should be removed from the cache when there is no more space to accommodate a new item.

In a mobile computing environment the solutions proposed for traditional distributed systems cannot be applied because they generate high network traffic and too much power consumption.

In this paper we present and classify different approaches for database cache management in mobile computing. A contribution of this paper is the taxonomy proposed for the problem and a comparative review of the work done in this area.

The remaining of this paper is organized as follows. In sections 2, 3 and 4 we propose and describe a taxonomy for cache management in mobile computing. Finally, in section 5 we discuss critical issues related to cache management in mobile databases.

2 TAXONOMY

Figure 1 illustrates the taxonomy for the cache management problem in mobile database proposed in this paper. We begin our classification considering the client's cache content. This way, three main categories can be identified: physical data storage, logical data storage and other storage forms.

Manica H. and S. de Camargo M. (2004).
CACHING ESTRATEGIES FOR MOBILE DATABASES.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 545-551
DOI: 10.5220/0002616205450551

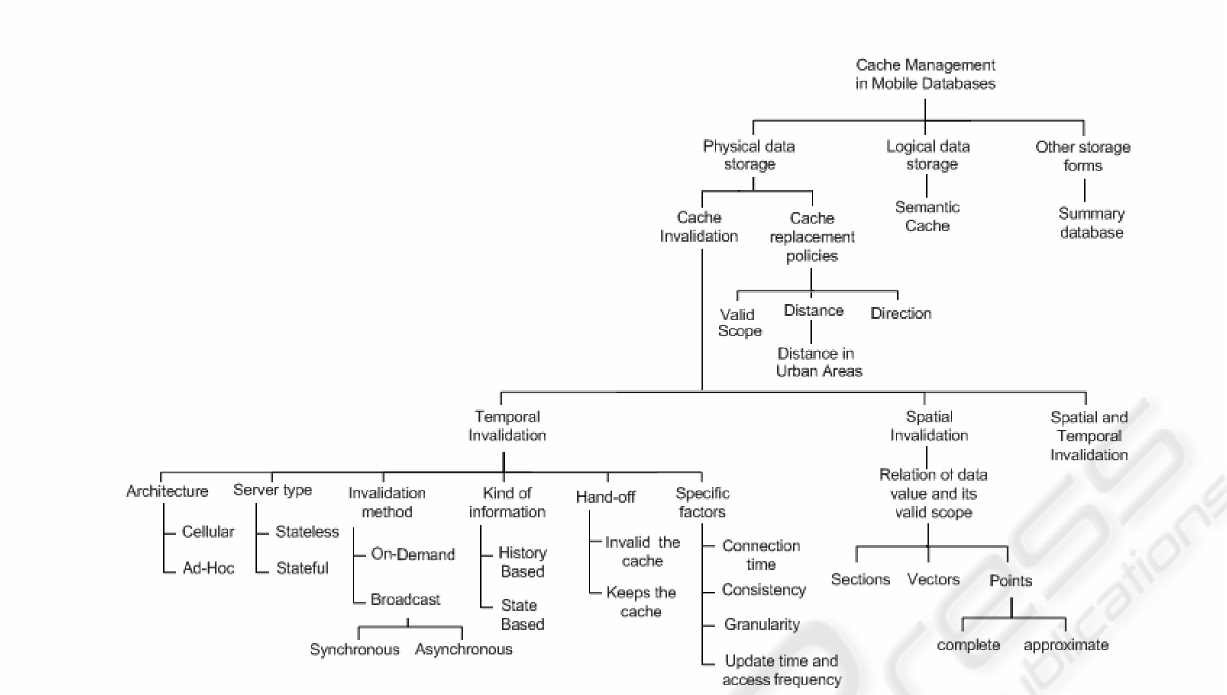


Figure 1: Taxonomy for cache management problem in mobile databases

In the physical data storage model the MU cache contents are copies (tuple or page) of data items from the server. Otherwise, in the logical data storage, arbitrary query answers are stored in the client’s cache.

Different from the physical model, the data is retrieved from the server using queries. This requires more processing capability at the server, but only the required data is transmitted over the wireless link.

The remaining of this section describes the cache invalidation and replacement in the physical data storage model. Next two sections will describe the logical data storage model and other storage forms.

2.1 Cache invalidation

Another classification for cache invalidation is found in (Tan et al., 2001). Our taxonomy considers that the cached data can become invalid either because of data updates in the server (temporal invalidation) or due to the client movement to another location area (spatial invalidation).

2.1.1 Temporal invalidation

In (Barbara & Imielinsk, 1994) the authors presented the first models introducing cache inconsistency techniques for mobile computing.

They proposed some interesting strategies based on invalidation reports (IR) that encouraged the appearing of new ones [(Hu & Lee, 1998), (Yen et al., 2000), (Kahol et al., 2001), (Cao, 2002)].

Next, we introduce the temporal invalidation parameters considered in our taxonomy proposal.

Architecture. Cellular networks are composed of a fixed component (where servers are located), by several cells covered by mobile support stations - (MSS) and by mobile units (MUs). The communication channel is divided into downlink (server to client) and uplink (clients to server). Another architecture is the ad-hoc network where computing devices are able to change information directly, without MSS help.

Server type. The server can be either statefull or stateless. A statefull server knows what MUs reside in its cell as well as their caches states. A stateless server has no information about MUs.

Invalidation method. The invalidation method refers to the way in which the server is going to keep the client up-to-date. Two types are commonly used: broadcast or on-demand. In the broadcast, the server sends IRs to clients by the downlink channel. The clients “hear” this channel and filter the needed data, without using the uplink channel. In the on-demand method, the client asks the server to check their caches validity using the uplink channel. The IRs can be sent through broadcast synchronously or asynchronously. In the asynchronous method, the IRs are sent immediately after changes on data items have occurred. In the synchronous method IRs are broadcasted periodically.

Kind of information. The information sent in the IR can be values of the data that were changed since the last report (state based) or just information about the modified items, e.g. their identifiers (history based).

Handoff. Handoff is a process that occurs when a MU crosses the boundary from a cell to another. In a handoff, the data cache can be kept or completely invalidated.

Some strategies may be more specific and consider characteristics such as the connection time, the cache consistency or the granularity level.

Next we briefly describe some relevant temporal invalidation strategies proposed in the literature. Table1 describes the invalidation parameters of each strategy.

Barbara and Imielinsk (1994) proposed diferent techniques named Broadcasting Timestamps (TS), Adaptive Invalidation Reports (AIR), Amnestic Terminals (AT), Signatures (SIG) and Quasi-copies.

In the TS strategy the server broadcasts IRs with the timestamps of the data items that have changed in the last w seconds. If the cache item has a timestamp smaller than the one in the IR, then it must be updated.

The method AIR proposes to extend the window size w for items requested frequently by MU that remains disconnected for a long time. They also mention that TS should have the window size w variable depending on the data item.

In AT strategy the server informs the data items identifiers that have changed since the last IR. The MU compares the items in its cache with those in the IR. If the item is in the report, then the MU drops it from its cache, else it considers the cache as valid.

Signatures are *checksums* computed over the data items values. This technique compares data files and checks their differences. The MUs subscribe to the items of their interests. The items that are not in the subscribed sets are considered equal to the ones in the IR that are being broadcasted.

The quasi-copies technique allows different values from the server in a controlled mode. For instance, a MU stores product prices, it is acceptable to use values that are not updated since they differ less than 0.5 % of the value stored in the server.

Some data can have their values often changed, such as the weather forecast. Others can have their values sporadically updated as a client phone. Considering this fact, the proposal in (Yen et al., 2000) associates an absolute validity interval (AVI) for each cached item. The cached item is invalidated if the access time is greater than the last update time by its AVI.

In the lazy pull-based model (Chan et al., 1998) each client is responsible for invalidating its own cache items. The update reports are sent to clients on demand and the cached items are only validated when they are accessed. First, the query is sent to the server that verifies the validity of the client’s cache. If necessary, the server sends back updated data items or those that are not in the client's cache.

Different from TS and AT, the strategy Asynchronous and Statefull (AS) is based on asynchronous IR and statefull servers (Kahol et al., 2001). Each MU maintains its own Home Location Cache (HLC) to deal with disconnections. Invalidation messages from servers are stored in the HLC while the MU is disconnected. When the MU is reconnected, the invalidation messages stored are delivered to it.

The Bit-Sequence (BS) invalidation strategy (Jing et al., 1997) uses a bit-sequence (or bit-vector) to refer to data items in the report. That is, each bit represents a data item in the database. The value 1 (one) means that the given item was updated in the server, and 0 (zero) means that it did not have any update. This strategy can use grouping methods to decrease the report size.

In the Local Optimal Strategy (LOP), (Hara, 2002), a MU is able to access data items from other MU cache. When a MU requests a data item, it verifies if the data is in its local cache. If it is not the case, the MU obtains the data item from the broadcast channel or from other MU cache.

Global Optimal Strategy (GOP). The bases of GOP (Hara, 2002) strategy are the same as the previous LOP. The main difference is that GOP strategy replaces its cache considering the share of cached items among all connected MU.

Stable Group Optimal (SOP). The application framework of SOP strategy (Hara, 2002) is the same as in LOP and GOP. Additionally, SOP proposes that data items are cooperatively cached in stable groups of MUs. The invalidation is done considering the data cache of all MUs connected to the group.

The Counter-based (CB) model (Cao, 2002) maintains a counter to every data item. Using counters, the server is able to know which data items are more frequently accessed (hot data items). The server keeps the identification of each client (statefull server). When a handoff takes place, the client should notify its old position to the new MSS, so that it can recover information about the client.

The model Scalable Asynchronous Cache Consistency Scheme (SACCS), (Wang et al., 2003), requires that MSS identify its database objects that are valid in the MU caches, instead of recognize all the objects of all the MUs. It introduces three main characteristics: the use of a flag at server and MU’s cache; use of an identifier for each entry in the MU’s cache after its invalidation; and all valid entries of MU’s cache are configured with an uncertain state when the MU is reconnected to the system.

Table1: Characteristics of temporal cache invalidation strategies

Strategies Characteristics	Architecture	Server Type	Invalidation Method	Information Type	Hand-Off	Disconnection Time	Update Time And Access Frequency	Consistency	Granularity
TS, AIR AND AT	cellular	stateless	broadcast synchronous	history based	invalid the cache	yes	n.v.	n.v.	mentioned as future work
SIG	cellular	stateless	broadcast synchronous	state based	invalid the cache	yes	mentioned as future work	n.v.	mentioned as future work
Quasi-Copies	cellular	stateless	broadcast synchronous	history based	invalid the cache	n.v.	n.v.	relax the consistency	n.v.
IAVI	cellular	stateless	broadcast periodic	history based	n.v.	n.v.	yes	considers a reasonable average	n.v.
Lazy Pull-Based	point-to-point	n.v.	on-demand	state based	n.v.	n.v.	yes	n.v.	considers tree data granularity levels
AS	cellular	statefull	broadcast asynchronous call-back	history based	keeps the cache	yes	n.v.	n.v.	n.v.
BS	cellular	stateless	broadcast periodic	history based	n.v.	n.v.	n.v.	n.v.	considers a rough bit granularity
LOP, GOP SOP	ad-hoc	stateless	broadcast synchronous	state based	n.v.	n.v.	yes	n.v.	n.v.
CB	cellular	statefull	broadcast	state based	keeps the cache	n.v.	n.v.	n.v.	n.v.
SACCS	cellular	stateless and statefull	broadcast	state based	mark the data items with a unstable state	n.v.	n.v.	n.v.	n.v.

n.v. = no verified

2.1.2 Spatial Invalidation

Spatial invalidation occurs when data values stored in cache become invalid because of the client movement to a new location area.

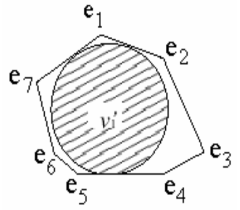
The maintenance of a valid cache when the clients move is called location-dependent cache invalidation and a data item can have different value depending on its location (Zheng et al., 2002).

The main factor considered in the spatial invalidation is the valid scope (or valid area). The valid scope of an item value is defined as the set of cells in which the item value is valid. There are different forms to relate the data value with its valid area.

The system can store *points* that represent the geographical region in which the data is valid. It can store either all the points or in an approximate form, only a part of the geographical region. Other form is to use *vectors* to save information about the region that data is valid. In another form, the system divides the database in logic sections according to the region in which the data is valid.

Next, we present some important spatial invalidation strategies proposed in the literature.

The strategy Polygonal Endpoints - PE (Zheng et al., 2002) stores all endpoints of the polygon representing the valid scope of a data item stored. When the number of endpoints is large, this technique will consume a large portion of the wireless bandwidth and space for caching the valid scope in the client. The advantage is the complete knowledge of the valid scopes.



v'_1=valid area circle

Figure 2: Possible valid areas. From (Zheng et al. 2002)

Zheng et al. (2002) proposed the utilization of an approximate circle (AC) inserted inside the original polygon (v' in figure 2). Thus, the valid area will be the approximate area defined by the center and radius of the circle. A problem occurs when the polygon shape is thin and long. In this case, the approximating error will be high and the cache can consider valid data as invalid if the query is outside the circle. Caching-Efficiency-Based (CEB) also proposed in (Zheng et al., 2002), is a generic method for balancing the overhead and the precision of the valid scopes to be attached.

The method Bit Vector with Compression BVC (Xu et al., 1999) considers that each cell has an identification (ID) and it uses a bit vector to record scope information. The bit vector length is equal to the number of cells in the system and all data in cache is associated to a bit in the vector. This way, the value 1 (one) means that the data item is valid in the cell and 0 (zero) that it is invalid.

Grouped Bit Vector with Compression (GBVC), (Xu et al., 1999), proposes to store information about cells that are adjacent or near MU current location. The model proposes the division of the wide geographical area of the system into groups and intra-groups. The cell ID consists of two parts: group ID and subgroups ID.

Another model also proposed in (Xu et al., 1999) is the Implicit Scope Information (ISI). This model divides the database into multiple logic sections. Data items with the same valid area are placed at the same section. The data item in cache will have the format {D_i, SDN_i and SN_i}, where D_i is the value of the data item, SDN_i is the section number, and SN_i the data number inside the section (scope number).

2.1.3 Spatial and temporal invalidation

In the literature, there is little work considering both the spatial and temporal invalidation. The authors in (Xu et al., 2003) present a performance study of the strategies BVC, GBVC and ISI in a scenario where temporal and spatial updates coexist.

2.2 Cache replacement policies

In location-dependent data services (LDD), the cache replacement policy must consider other factors besides the access probability such as: movement, direction, speed, etc. The factors considered in our taxonomy are valid scope, distance and direction.

Valid Scope Area. Valid scope is the geometric area in which the data value is valid. A common way to perform location-dependent cache invalidation is to attach the valid scopes to the data values returned to the client (Xu et al., 1999).

Distance. In LDD, the server answers queries according to the client's location. When the valid scope of a data value is far away from the current client's location, this data will have a lower chance to become useful.

Distance in Urban Areas. The computation of the distance between client's location and the data valid area can change according to the kind of application. In a rural zone for example, we can use the Euclidean distance $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. However, in an urban area this formula is not suitable because the MU can move through the streets with buildings or other obstacles.

Direction. The direction can be used first to eliminate from the cache the data that are in the opposite direction of the client's movement.

Next, we describe some proposed policies for cache replacement.

The Probability Area (PA), (Zheng et al., 2002) defines the data items to be replaced according to a cost function defined as the product of the access probability of a data item and its attached valid scope. The cost function of a value j of an item i is $C_{i,j} = Pi \cdot A(v_{i,j})$, where Pi is the access probability of the item i and A(v'$_{i,j}$) is the valid scope for a value j of an item i. When the data replacement is carried out, this policy selects the data with the least cost (s).

In the Probability Area Inverse Distance (PAID) policy (Zheng et al., 2002), the cost function of a value j of an item i is given by C $_{i,j}$ = Pi . A(v'$_{i,j}$) / D(v'$_{i,j}$) where Pi and A(v'$_{i,j}$) is defined in the same way as above, and D(v'$_{i,j}$) is the distance between the current location and the valid area v'$_{i,j}$. When data replacement is carried out, PAID ejects the data value (s) with the least cost (s).

To consider the movement direction, the authors in (Zheng et al., 2002) take extensions of the model PAID: PAID-U (Probability Area Inverse Distance - Unidirectional) and PAID-D (Probability Area Inverse Distance - Directional). In PAID-D, the distance is calculated considering the client's current direction of movement. PAID-D keeps the data that are in the direction of the client's movement. On the other hand, in PAID-U the distance is computed regardless of the current direction of the client's movement.

The Manhattan Distance (MD) policy (Jung et al., 2002) is suitable for location dependent queries in urban area. The distances in urban zones are given by |(x1-x2)|+|(y1-y2)|. The proposed algorithm computes the Cache Replacement Score (CRS) based on the MD computation. It chooses the victims according to CRS by the current location of the MU. Thus, the victims who are farthest from the MU will be replaced first.

3 LOGICAL DATA STORAGE

The semantic cache model (SC) is an attractive approach for mobile computing. The idea is to maintain in the client's cache both the semantic descriptions and associated answers for previous queries. The SC utilizes the semantic information to organize and to manage the client's cache.

The query processing uses the semantic descriptions to determine which data is available in the cache and which ones will have to be requested to the server. The semantic description is also utilized in the definition of the cache replacement policy, not requiring any additional attached information to each tuple as in traditional cache management systems.

In a basic version of this approach, the semantic segment is represented by the set (S_R, S_A, S_P, and S_C), where S_R and S_A define the relation and the involved attributes. S_P denotes the selection condition and S_C represents the result (pointer for the result pages).

A segment S contains all or part of the result of a query Q. Thus, Q is divided into two parts (query trimming). The first one is part of the query result that is in S, called probe query. The second one is the part that could not be found in S, called reminder query.

After the query splitting by the first segment, the next candidate segment will divide again the remaining query. This process continues until there are no more candidate segments or the reminder query finishes. At the end, if the reminder query is not empty it is sent to the database server to be computed. When the server returns the answer, the whole result is composed by all probe and reminder queries.

The affinity refers to the kind of relationship between the data items in the cache. This relationship can be temporal (temporal locality) or semantic (semantic locality). Temporal locality is the property that items that have been referenced recently are likely to be referenced again in the near future. Semantic Locality is the property that if an item has been referenced, other items with the same semantic function (for instance, the nearest) are also likely to be referenced.

In the following, we introduce some models for replacement in semantic cache for mobile computing.

Dynamic Least Recent Used (D-LRU), (Ren & Dunham, 1998), is an adaptation of the traditional LRU approach. Considering that the segments can be shrunk or enlarged, the algorithm D-LRU only updates the value of the timestamp when a new segment is created or when a new element is added. If it is shrunk, an old timestamp is kept.

The two-level LRU model (Ren & Dunham, 1999) uses an additional structure called clusters that are groups of semantically related segments. The first level LRU policy is carried out at a cluster level. The cluster with the oldest timestamp is selected as candidate to be further examined. Next, for all the segments in this cluster, a second level LRU is run and replaces the items that are least recently accessed until the cache space is enough to hold the new query.

Furthest Away Replacement (FAR), (Ren & Dunham, 2000), is a solution for SC replacement in LDD. This policy classifies the segments in two sets: the first one with segments that are in the direction of the movement, and the second with segments that are out of the direction. The victims are always selected from out-direction set. When the out-direction set is empty, the most distant segments of the in-direction set are replaced.

4 OTHER WAYS OF STORAGE

An example of other ways of storage data is the summary database. The main goal of the summary database model (Madria & Roddick, 1998) is to increase the data availability through the construction of a summary of the main database (complete in the MSS) that will be stored in the client's cache.

Thus, the mobile client can process a query using summarized data in cache returning approximate results while MSS returns an exact result. To compute the database summary, it uses the hierarchy concept, generally defined by database administrator.

This model is able to provide several approximated levels of answers for queries carried out by the MU, using data stored locally in the client or remotely in the server. The MU can stay disconnected using a portion of the local database. In long time connections, the local data can be updated, avoiding a high degree of outdated data.

5 FINAL REMARKS

In this paper, we have discussed cache invalidation and replacement issues in the context of mobile computing. In summary, we have presented several solutions proposed in the literature and we have also proposed a taxonomy for this problem.

Regarding physical storage, we have presented the main areas in cache management: query invalidation caused by both temporal- and location-dependent updates, and cache replacement considering the valid scope for the data items. For the logical storage, we have presented the semantic cache model and finally, as other storages formats, we have presented the summary database model.

This paper provides a basis for identifying strengths and weaknesses of individual methodologies, as well as general guidelines for future improvements and extensions. The semantic cache model for LDD opens up a new dimension of research in mobile computing. Some improvement can still be made on the proposed cache replacement schemes.

Since mobile clients may have different movement patterns, adaptive techniques can be developed. Considering that clients can move, but also stay temporarily fixed, we will continue our research work looking for a good proposal on semantic cache replacement policy that considers user's behavior to decide the best way to replace the data in the client's cache.

Furthermore, an important and challenging area that is in need of good solutions is the data management in independent or ad-hoc network. In (Chiasserini et al., 2003) is treated the cache placement problem, but with static topology. There are few proposals for cache management for this kind of mobile architecture (Tassiulas & Su, 1997).

REFERENCES

Barbara, D. and Imielinski, T., 1994. Sleepers and Workaholics: Caching Strategies in Mobile Environments. In *Proceedings of ACM SIGMOD*. May 1994.

Chiasserini, C. F.; Nuggehalli, P. and Srinivasan, V., 2003 Energy-Efficient Caching Strategies in Ad-Hoc Wireless Networks. In *Proceedings of MobiHoc'03*.

Cao, G., 2002. A Scalable Low-latency Cache Invalidation Strategy for Mobile Environments. In *IEEE Transactions on Knowledge and Data Engineering*.

Chan, B. Y.; Si, A. and Leong, H. V., 1998. Cache Management for Mobile Databases: Design and Evaluation. In *Proceedings of the 14th International Conference on Data Engineering*.

Gray, C. G. and Cheriton, D. R., 1989. Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency. In *Proceedings of SOSP'89*.

Hara, T., 2002. Cooperative Caching by Mobile Clients in Push-based Information Systems. In *Proceedings of CIKM'02*, November 4-9, Mc Lean, Virginia, USA.

Hu, Q.L. and Lee, D.L., 1998. Cache Algorithms Based on Adaptive Invalidation Reports for Mobile Environments. In C*luster Computing, 1 (1): 39-48*, February 1998.

Jing, J.; Elmagarmid, A. K.; Helal, A. and Alonso, R., 1997. Bit-sequences: A new cache invalidation method in mobile environments. In *ACM/MONET*, 2(2): 115-127.

Jung, Y. Y.; Lee, J. and Kim, K., 2002. Broadcasting and Caching Policies for Location-Dependent Queries in Urban Areas. *WMC'2002*. Georgia, USA.

Kahol, S. K.; Gupta, S. K. S. and Srimani, P. K., 2001. A strategy to Manage Cache Consistency in a Distributed Mobile Wireless Environment. *IEEE Transactions on Parallel and Distributed Systems (TPDS'2001)*.

Madria, S. K. and Roddick, M. M., 1998. A Query Processing Model for Mobile Computing using Concept Hierarchies and Summary Databases. *The 5th International Conference of Foundations of Data Organization*, Kobe Japan, November 12-13, 1998.

Ren, Q. and Dunham, M. H., 1998. Semantic Caching and Query Processing. *Technical Report 98-CSE-04. Department of Computer Science and Engineering*. Southern Methodist University, Dallas, May, 1998.

Ren, Q. and Dunham, M. H., 1999. Using Clustering for Effective Management of a Semantic Cache in Mobile Computing. In *ACM MobiDE*, Seattle – USA, 1999.

Ren, Q. and Dunham, M. H., 2000. Using Semantic Caching to Manage Location Dependent Data in Mobile Computing. In *Proceedings of Mobicom'2000*.

Tan, K. L.; Cai, J. and Ooi, B. C., 2001. An Evaluation of Cache Invalidation Strategies in Wireless Environments. *IEEE (TPDS)*, 12(08): 789-807.

Tassiulas, L. and Su, C. J., 1997. Optimal memory management strategies for a mobile user in broadcast data delivery system. IEEE Journal on Selected Areas in Communications (JSAC), 15 (7): 1226-1238.

Wang, Z.; Das, S.; Che, H. and Kumar, M., 2003. SACCS: Scalable Asynchronous Cache Consistency Scheme for Mobile Environments. In *(ICDCSW'03) - 23rd International Conference on Distributed Computing Systems Workshops*, pp. 797.

Xu, J.; Tang, X. and Lee, D. L., 2003. Performance Analysis of Location-Dependent Cache Invalidation Schemes for Mobile Environments. *TKDE* 15 (2) 474-488, 2003.

Xu, J.; Tang, X.; Lee, D. L. and Hu, Q., 1999. Cache Coherency in Location-Dependent Information Services for Mobile Environment. In *Proc. the 1st Int. Conf. on Mobile Data Access (MDA'99)*, Hong Kong, Springer-Verlag LNCS, vol. 1748, pp. 182-193.

Yen, J.; Chan, E.; Lam, K. Y. and Leung, H. W., 2000. An Adaptive AVI-Based Cache Invalidation Scheme for Mobile Computing Systems. In *11th International Workshop on Database and Expert Systems Applications - DEXA'2000*.

Zheng, B. ; Xu, J. and Lee, D. L., 2002. Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments. In *IEEE Trans. on Computers, Special Issue on Database Management and Mobile Computing*, 51(10): 1141-1153, October 2002.