A Location Dependent Semantic Cache Replacement Strategy in Mobile Environment

Kahkashan Tabassum, Mahmood Quadri Syed, and A. Damodaram

kahkashan@mjcollege.ac.in, syedmahmood.q@gmail.com, adamodaram@jntuh.ac.in

Abstract. Mobile computing is developing fast and one of its major services is location dependent information services (LDIS).The dependence of the results of a query on the present location of the mobile user leads to such services. The query is called Location Dependent Query and the resultant data is called Location Dependent Data (LDD).The caching scheme often used in these services is semantic caching where information about data is stored along with data in cache. In this paper, we have added a new dimension, segment frequency (S_F) to the Semantic segment. The cache replacement policy takes this dimension into consideration when replacing the cache. The prediction algorithm Enhanced RBFNN (ERBFNN) takes the future location of neighbors into account. The existing FAR algorithm is modified taking into account the new dimension to replace items from cache. The proposed system is called Enhanced RBF-FAR (ERBF-FAR) algorithm. The experimental results show that the proposed system performs better and yields better results.

Keywords: Mobile Computing, LDIS, LDQ, LDD, Semantic Caching, FAR, ERBFNN.

1 Introduction

The development of high speed wireless network and the increasing use of portable wireless devices have led to the development of mobile computing. The ability to move and know your \own location has given rise to services known as Location Dependent Information Services (LDIS).A Location Dependent Data (LDD) is the data whose value depends on the current location of the mobile user [1]. The queries used to process these kind of data is known as Location Dependent Query (LDQ).Location Dependent Querying is gaining increasing popularity in mobile computing systems. Mobile users in wireless communications face several difficulties like low bandwidth, frequent network disconnection, etc. So, Data caching is needed [2]. When the mobile user cache gets full then data items from cache has to be removed to accommodate new item [3]. This is known as cache replacement [4].The scheme that fits exactly for the purpose of Location dependent Querying is the Semantic caching scheme. The basic idea of semantic caching is that the information about the data should be stored along with the data in the cache [3]. For improved performances, several semantic caching models have been proposed. Whenever a new Location dependent query is generated, the system checks whether it can be

N. Meghanathan et al. (Eds.): CCSIT 2012, Part I, LNICST 84, pp. 213–224, 2012.

responded by the information stored in cache. If yes, then the answer is then and there. If no, then if the query is query is partially answerable then the only a trimmed part of query is sent to the server. Semantic caching this way helps to reduce the network load by decreasing the amount of data that is transferred over the network.

In this paper, we study a semantic caching model. The definition of semantic segment that was proposed previously is enhanced to accommodate a new dimension Segment Frequency (S_f).This factors plays an important role when the decision has to be taken for which item has to be replaced in cache to accommodate a new item. We also study the existing cache replacement polices like Furthest Away Replacement (FAR) where the data item that is furthest is away from cache is removed first form cache. This cache replacement policy also predicts the clients future movement based on velocity. Another existing cache replacement policy that improves upon FAR known as RBF-FAR is also studied which predicts the client's future location using a RBF Network [11]. We then Propose a new scheme which an improvement on existing RBF-FAR scheme known as Enhanced RBF-FAR (ERBF-FAR).The proposed scheme improves upon the existing scheme where the cache replacement policy and the prediction architecture has been improved to provide better performance. We then simulate the existing system with proposed system and produce results.

2 Literature Survey

In this section, we first introduce the semantic caching model. This model uses the select and projects queries on a Database D, which has Relations R1, R2, . . . , Rn, i.e., D = {Ri,1 <= i <= n} [11].An Attribute set A is also defined where A= UARi,1 <= i <= n [11].With this, We define The term semantic segment and query.

Semantic Segment. Given a database D = {Ri,1 <= i <= n}and its attribute set A= UARi,1 <= i <= n , a semantic segment, S [11], is a tuple <SR,SA,SP,Sm,SC,ST>,where $Sc = \Pi SA \sigma Sp(SR)$, SR ε D , SA ε ASR and SP indicates the select condition that the tuple in S satisfy, SC represents the actual content of S. The Sm represents the MU user coordinate that helps to predict future location of moving objects and used to new replacement policy. ST is the timestamp for semantic segment.

Query. A Query Q is a semantic segment, <Q_R,Q_A,Q_P,Q_m,Q_C> [11].The Parameter S_T is removed as this parameter is only used for taking replacement decision. The storage of segments in the memory is done by a technique called paging [1]. Every page has a segment which connects to other pages and has a variable which holds the address and refers to the starting first page. An index is used to structure the cache memory in an organized manner which maintains the semantic descriptions and storage information for each segment in the cache. An example structure of index can be shown by taking two relations: Hospital (Hno, Hname, Hx, Hy), Petrol Pump ((Pno, Pname, Px, Py). We assume that the mobile user gives queries at different locations on his way. The current location of mobile user is designated using M(x, y).we issue two queries Q1, Q2 respectively,

Q1 - MU (30, 40): Show all hospitals within 25 km.
Q2 - MU (10, 30): Show all Petrol pumps within 7 km.

Table 1. An Example of semantic cache index

S	SR	SP	Sm	Sc	ST
S1	Hospital	(MUx-25<=Hx<=MUx+25)^(MUy-25<=Hy<=MUy+25)	(30,40)	2	T1
S2	Petrol Pumps	(MUx-7<=Px<=MUx+7)^(MUy-7<=Py<=MUy+7)	(10,30)	5	T2

Query Processing. Whenever a query is issued, we first check whether it can be answered by the cache. If yes, then the results are then and there. If the query can only be partially answered, we trim the original query by removing the presently answered parts and send it to the server for processing. The issue related processing query is the development of semantic segment. To prevent repeated data we cannot keep S and Q in the cache. Whenever a query is issued, if it overlaps a semantic segment in cache different techniques can be used to evaluate them. There no coalescing approach generates three disjoint parts: (Q ∩ S), (S ^ ┐Q) and (┐S ^ Q*)* [9].Problem with this approach is that it may result in a many smaller segments. Complete coalescing approach is better for small queries but not for replacement [5]. The “partial coalescence in query” technique is the best [3]. The segment is decomposed in two parts: the overlapped part and the no overlapped part [6]. In this paper, we use the “partial coalescence in query” technique [5].

3 Existing System

In this section we will discuss the existing replacement policies and the existing prediction architecture. Furthest Away Replacement (FAR) policy is an existing policy which replaces the items which is furthest away from mobile client’s present location. It also predicts the mobile client’s future location using velocity. However, the FAR has its own problems which can be demonstrated by the following figure.

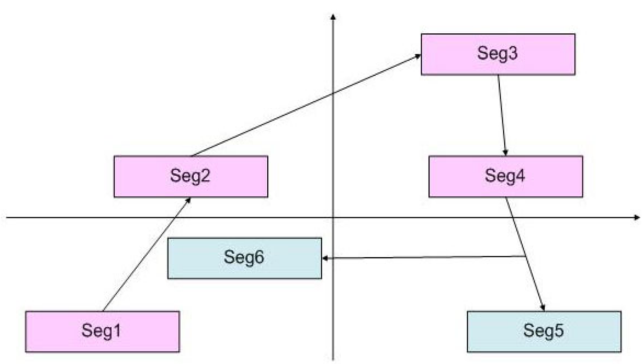

Fig. 1. Problems with FAR

In the figure above, Seg1, Seg 2, Seg 3, Seg 4, are semantic segments. Seg 5 and Seg 6 are assuming to be future semantic segments. We assume that, the Present timestamp is T4 of Seg 4, and T1<T2<T3<T4. Seg 5 or Seg 6 will be the semantic segment accessed if a new query is issued. When FAR algorithm is used, in Seg 4, the result that we get should be stored in cache but the space is not enough to store the result. Hence we have to remove an item from cache to accommodate the new item. Using FAR algorithm, the future location would be Seg 5 based on current velocity of Seg 4. So FAR replacement policy will remove Seg 1. But, if the mobile user changes its direction to Seg 6 half way then the next query will be generated on Seg 6 rather than Seg 5.Hence the Seg 1 which was removed was useful and should not have been removed. This was the result of inaccurate prediction by FAR algorithm. To overcome the problems with FAR, A new prediction model was developed which used RBF-Network to predict mobile user's future location [7]. The RBFNN is self learning model which takes the current location of mobile user M(x, y) using the S_m of Semantic segment and the timestamp S_t as input to the prediction architecture which gives the future location of mobile user $M_{fl}(x, y)$.The algorithm was called RBF-FAR [11].it uses RBF-Network to predict future location and based on that FAR replacement policy is used to replace the items in cache. The prediction architecture can be illustrated through the figure below.

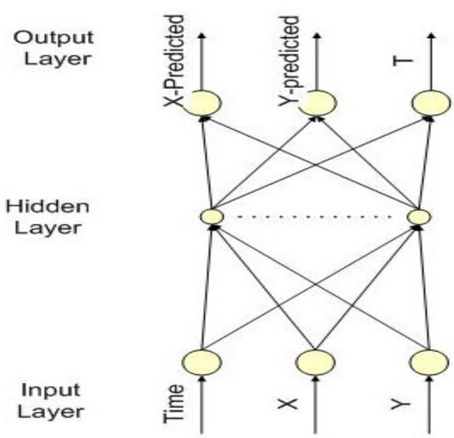

Fig. 2. Prediction Architecture

The whole description of how the RBF-FAR Replacement policy works is given by the following algorithm [11].

```
Algorithm RBF-FAR(C, M)
{
In-Direction ←NULL;
Out-Direction←NULL;
Call RBFNN to predicate location;
Mfl= (x-predicated, y-predicated);
for every segment seg in C
```

```
{
if Distance(segL, MfL) ≤ Distance(segL, ML)
then In-Direction ←In-Direction +{seg};
else Out-Direction ← Out-Direction +{seg};
}
while ( Out-Direction != Empty )
{
seg ← the segment in Out-Direction which is the furthest from M;
discard seg from C;
remove seg from Out-Direction;
add free space;
if ( free space is enough )
return (Success);
}
while ( In-Direction != Empty )
{
seg ← the segment in In-Direction which is the furthest from M;
discard seg from C;
remove seg from In-Direction;
add free space;
if ( free space is enough )
return (Success);
}
return (Fail);
}
```

The RBF-FAR algorithm should be trained for good performance. However if the training data is not sufficient then the RBFNN will give a lower performance [11].

4 Proposed ERBF-FAR Scheme

The existing RBF-FAR scheme needs to be trained for good performances and requires good amount of training data to perform well. The proposed ERBF-FAR scheme extends the RBF-FAR scheme for acquiring good performance. A new dimension S_f is added to the existing semantic segment which is used for cache replacement decision. The future location of the neighboring mobile users is given as input to the RBF-Network which improves the quality of prediction. The improvements in proposed scheme are discussed one by one in this section.

Proposed Semantic Segment. Given a database D = {Ri,1<=i<=n}and its attribute set A= UARi,1 <=i<=n, a semantic segment, S, is a tuple <SR,SA,SP,Sm,SC,ST,**SF**>, where $Sc = \Pi SA \sigma Sp(SR)$, SR ε D , SA ε ASR and SP indicates the select condition that the tuple in S satisfy, SC represents the actual content of S. The Sm represents the MU user coordinate that helps to predict future locations of moving objects and used to new replacement policy. ST is the timestamp for semantic segment; **Sf** is the frequency of the semantic segment.

Proposed Replacement Policy. The cache replacement policy takes the newly proposed dimension S_f into consideration for replacing an item from cache. The S_f denotes number of time the semantic segment was accessed. Whenever a new query results are obtained then an item in the cache has to be replaced to accommodate new results if the cache is full. According to FAR the items that are furthest away from current location should be removed. However In the proposed scheme we take the average of value of S_f of all semantic segment called S_{favg}. When the cache replacement decision is to be taken then we remove only the segment which has S_f less than S_{favg} and furthest away from current location. If a data item is furthest away from current location but its s_f is greater than or equal to average S_f then its should be removed from cache. The immediate next segment which is less further but has S_f less than S_{favg} should be removed. If all the items in cache have S_f greater then s_{favg} then the item with lowest S_f value should be removed.

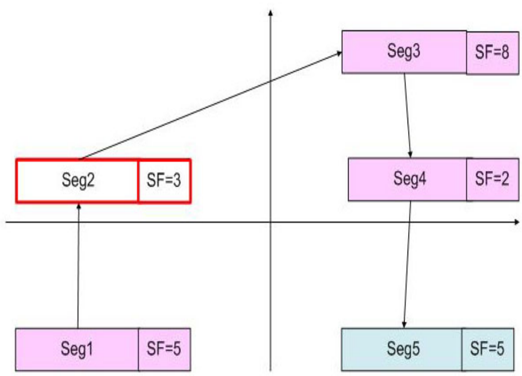

Fig. 3. ERBF-FAR Cache Replacement Scheme

In the above figure, a new query is issued at seg 4 and the results are obtained. We assume the cache is full so we need to replace an item from cache.seg 1 is the farthest away segment from seg 4 but it has S_f value 5 which is greater that average frequency 4.so the next farthest segment seg 2 is removed as it has S_f less than average frequency.

Prediction Architecture. The prediction architecture in the proposed system takes into account the future locations of neighboring mobile clients. The proposed prediction architecture is called Enhanced RBFNN (ERBFNN) which is an extension to RBFNN. The proposed prediction architecture will give improved performance. A new mobile user will not have any training data based on which prediction can be done. So training data can be taken from neighboring mobile clients who have good training sets and future locations can be accurately predicted.

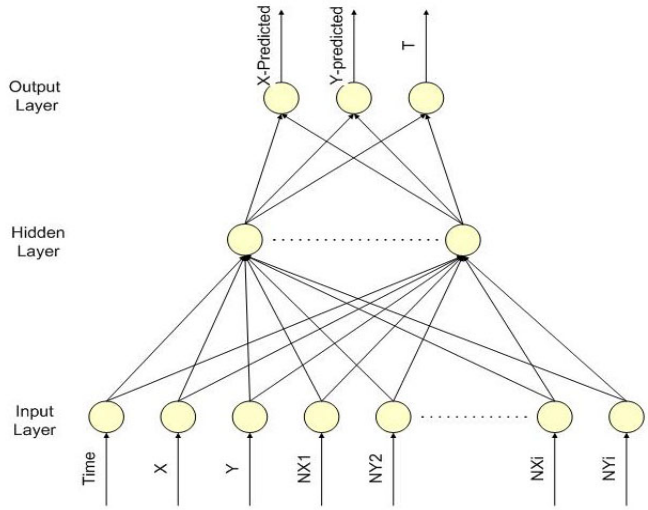

Fig. 4. Proposed Prediction Architecture

The whole description of how the ERBF-FAR Replacement policy works is given by the following algorithm.

```
Algorithm ERBF-FAR(C, M)
{
In-Direction ←NULL;
Out-Direction←NULL;
Call ERBFNN to predicate location;
Mfl= (x-predicated, y-predicated);
for every segment seg in C
{
if Distance(segL, MfL) ≤  Distance(segL, ML)
then In-Direction ←In-Direction +{seg};
else Out-Direction ← Out-Direction +{seg};
}
Favgi  = average Sf of all segments in In-Direction.
Favgo = average Sf of all segments in Out-Direction.
while ( Out-Direction != Empty )
{
seg ← the segment in Out-Direction which is
the furthest from M;
while ( Out-Direction != Empty )
{
if(seg.Sf<Favgo)
{
discard seg from C;
```

```
24: remove seg from Out-Direction;
25: add free space;
26: if ( free space is enough )
27: return (Success);
28: }
29: Else
30: {
31: Move to next furthest segment in Out-Direction
32: Repeat 19.
33: }
34: }
35: Seg ← segment with lowest Sf in Out-Direction.
36: Goto step 23
37: }
38: while ( In-Direction != Empty )
39: {
40: seg ← the segment in In-Direction furthest from M;
41: while ( In-Direction != Empty )
42: {
43: if(seg.Sf<Favgi)
44: {
45: discard seg from C;
46: remove seg from In-Direction;
47: add free space;
48: if ( free space is enough )
49: return (Success);
50: }
51: Else
52: {
53: Move to next furthest segment in In-Direction
54: Repeat 41.
55: }
56: }
57: Seg ← segment with lowest Sf .
58: Goto step 45
59: }
60: return (Fail);
61: }
```

In this algorithm, if the mobile user moves in a well known location the replacement technique has a good performance. If user M moves to a new area, the FAR and RBF-FAR will be low effective whereas ERBF-FAR is more effective as takes neighbors into consideration for future location prediction.

5 Implementation Results

The design of our simulation scenario consists of one server, one mobile user and a wireless channel between them to communicate. The mobile user issues queries and the server responds to the queries by maintaining a database containing information to serve the mobile user. We consider that the mobile client has one of the three cache replacement policies i.e., FAR, RBF-FAR and ERBF-FAR. In the mobile client simulation scenario, we have LDQ which is generated on the basis of workload, A Semantic cache manager which manages the semantic cache memory and a semantic cache query processor which processes the queries. The system parameters are displayed in the following table.

Table 2. System parameter

Parameter	Description	Value
MUProcessor	Mobile user CPU speed	500(Mips)
MUcach	Mobile client cache size(KB)	512
BW	Wireless channel bandwidth	32 K
LenPage	Size of the data page(bytes)	4096
MsgFC	Fixed part protocol cost to send/receive message	10000
MsgPB	Size-dependent part protocol cost to S/R message	500
AttrSel	The attributes to be queried	AX,AY
QSel	Query selection	4 sets

The working design of the simulation scenario made up of three relations. One relation is a 1000 set and other two relations are 2000 set. We take two important attributes AX and AY. AX is indexed and unclustered and AY is indexed and clustered. We use select queries to generate LDQ's. We use location as predicate. AttrSel is used to specify attributes upon which query is generated and QSel is used to specify the query. The queries are generated using five random sets consisting of 1000 LDQ's. In Each set, the starting 100 queries are used as warm-up data and the rest 900 queries are used as test data. The LDQ's are well defined. The average time and network load is tested to show the effectiveness and efficiency of the model.

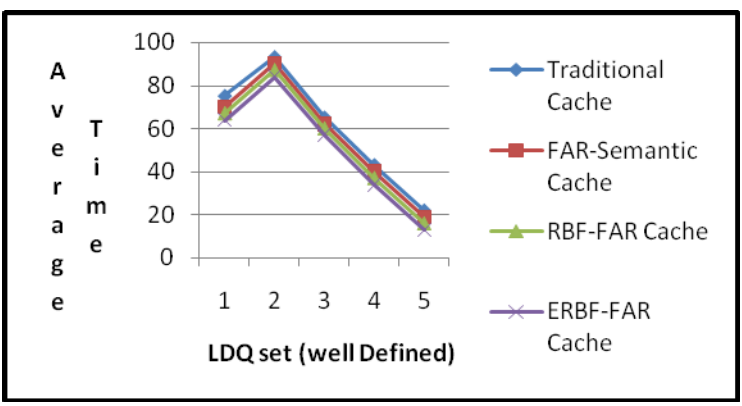

Fig. 5. Average response time vs well-defined paths

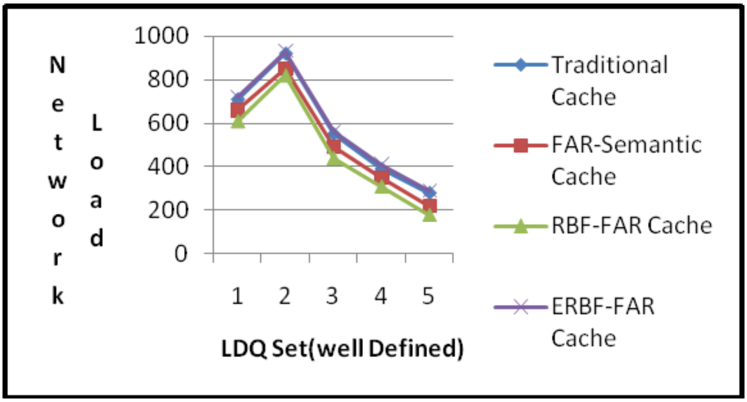

Fig. 6. Network-load vs. well-defined paths

The experimental results show that the semantic cache has less average response time than the traditional cache. The ERBF-FAR gives a good average response time performance than others. The network load has increased in the proposed system as the communication overhead has increased due to high communication with neighboring mobile clients for accessing their future locations. We also define LDQ's where mobile user is not well defined which means that the mobile user moves to a new location every time which it has previously not visited. The ERBF-FAR gives a good performance as it incorporates the future location of neighboring mobile users which increases the quality of future location prediction. The proposed scheme gives a better average response time than other schemes and the network load is increased due to communication overhead. But still it's sacrificed due good quality of prediction.

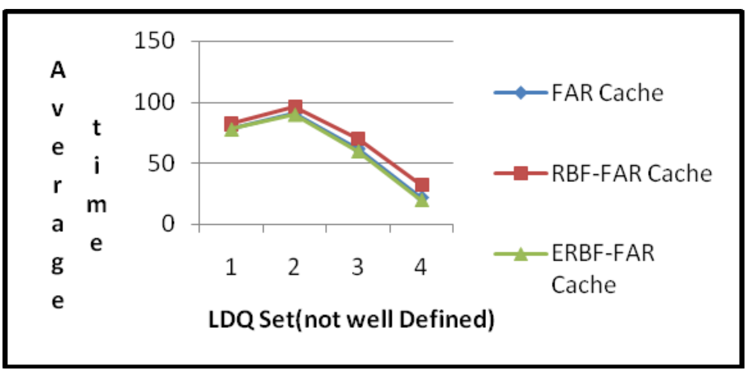

Fig. 7. Average time of response vs. poor-defined paths

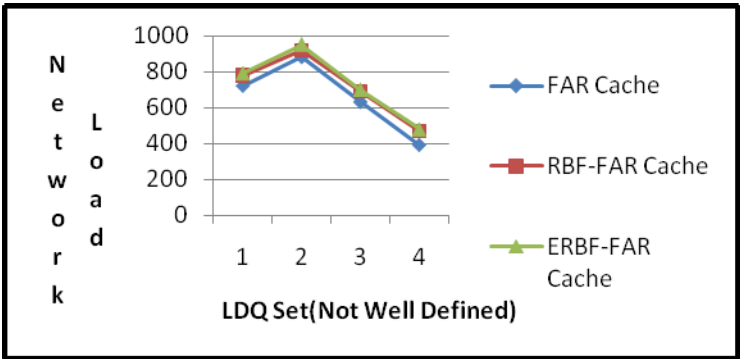

Fig. 8. Network-load vs. poor-defined paths

The main characteristics of a good cache replacement policy are that it will maintain a high cache hit ratio to improve the system performance. The ERBF-FAR shows an improved performance over other cache replacement policy and gives a high cache hit ratio. The addition of factor S_f to the semantic segment and its influence on the cache replacement decision has led to the increased cache hit ratio. The results are shown as following.

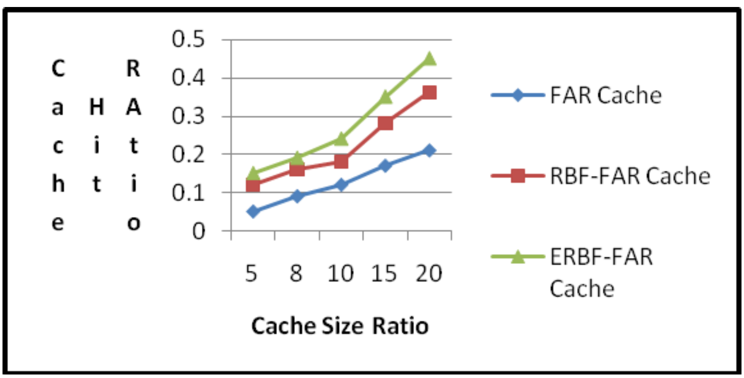

Fig. 9. Cache hit ratio vs. cache sizes

Hence the ERBF-FAR shows an improved performance and high cache hit ratio and is twice better than RBF-FAR and thrice better than FAR.

6 Conclusion

In this paper, we have discussed various cache management issues for location dependent data. We also defined the existing semantic segment, query and semantic segment index. The existing cache replacement policies were also discussed. The FAR cache replacement policy replaces the furthest away item and uses velocity for prediction of future location. The RBF-FAR uses FAR policy for replacement and RBF-Network for prediction of future location. The Proposed ERBF-FAR scheme proposes a new semantic segment which adds a new dimension Segment frequency (S_f) which is used for taking cache replacement decision. The prediction architecture is improved by adding future location of neighboring mobile users as input to the RBF-Network which gives an improved performance.

However, the introduction of future location of neighboring mobile users as increased the network load but has improved the quality of future location prediction. Hence, the Experimental results shows that, ERBF-FAR shows an improved performance over existing policies and gives better average response time and cache hit ratio.

References

1. Ren, Q., Dunham, M.H.: Using Semantic Caching to Manage Location Dependent Data in Mobile Computing. ACM (2000) 1-58113-197-6
2. Franlin, M.J.: Client Data caching. A Foundation for High Performance Object Database systems. Kluwer Academic (1996)
3. Dar, S., Franklin, M.J., Jonsson, B.T., Srivatava, D., Tan, M.: Semantic Data Caching and Replacement. In: Proc. VLDB Conf., pp. 330–341 (1996)
4. Ren, Q., Dunham, M.H.: Semantic Caching and Query Processing. IEEE Transactions on Knowledge & Data Engg. 15(1) (January 2003)
5. Dar, S., Franklin M.J., Johnson, B.T., Srivastav, D., Tan, M.: Semantic Data Caching and Replacement. In: Proc. - 22nd VLDB Conf., Mumbai, India (1996)
6. Manica, H., de Camargo, S.M.: A new Model for Location-Dependent Semantic Cache Based on Pre-Defined Regions. CLEI (2004)
7. Bilurkar, P., Rao, N.: Application of neural network techniques for location prediction in mobile networking. In: Proceedings of the 9th International Conference on Neural Information Processing, vol. 5, pp. 2157–2161
8. Wu, T.-T., Su, W.-Y., Zhou, X.-M.: Mobile Query Through Semantic Cache. Journal of Computer Research and Development 41(1)
9. Chen, C.M., Roussopoulos, N.: The Implementation and Performance Evaluation of the ADMS Query Optimizer: Integrating Query Result Caching and Matching. In: Jarke, M., Bubenko, J., Jeffery, K. (eds.) EDBT 1994. LNCS, vol. 779, pp. 323–336. Springer, Heidelberg (1994)
10. Gray, J.: The Benchmark Handbook. Morgan Kaufmann (1993)
11. Li, Z., He, P., Lei, M.: Research of Semantic Caching for LDQ in Mobile Network (2005)