韓國컴퓨터情報學會 論文誌
第17卷 第2號, 2012. 2.

2012-17-2-1-2

A mobile data caching synchronization strategy based on in-demand replacement priority

Jinhua Zhao*, Ying Xia*, Soon-Jo Lee***, Hae-Young Bae**

수요에 따른 교체 우선 순위 기반 모바일 데이터베이스 캐쉬 동기화 정책

조 진 화*, 하 영*, 이 순 조***, 배 해 영**

Abstract

Mobile data caching is usually used as an effective way to improve the speed of local transaction processing and reduce server load. In mobile database environment, due to its characters – low bandwidth, excessive latency and intermittent network, caching is especially crucial. A lot of mobile data caching strategies have been proposed to handle these problems over the last few years. However, with smart phone widely application these approaches cannot support vast data requirements efficiently. In this paper, to make full use of cache data, lower wireless transmission quantity and raise transaction success rate, we design a new mobile data caching synchronization strategy based on in-demand and replacement priority. We experimentally verify that our techniques significantly reduce quantity of wireless transmission and improve transaction success rate, especially when mobile client request a large amount of data.

▶ Keyword : mobile data caching, HDC, synchronization, priority

요 약

모바일 데이터 캐쉉 기법은 로컬 데이터 전송 과정에서 속도를 향상시키거나 서버의 오버로드를 줄이기 위한 효과적인 기법으로 많이 사용되었고 모바일 컴퓨팅 환경의 저전력, 접속지연 및 간헐적인 인터넷 연결 등의 제약 사항의 해결을 위해 캐싱 기법을 사용하는 다수의 연구가 진행되었다. 그러나 최근 스마트폰이 대량 보급되면서 서버 기

• 제1저자 : 조진화 • 교신저자 : 이순조
• 투고일 : 2011. 12. 12, 심사일 : 2011. 12. 29, 게재확정일 : 2012. 01. 12

* 중국 중경우전대학교 컴퓨터정보공학과(Dept. of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China),
** 인하대학교 컴퓨터정보공학과(Dept. of Computer & Information Engineering, Inha University),
*** 서원대학교 컴퓨터공학과(Dept. of Computer Engineering, Seowon University)

반 어플리케이션 등 다양한 어플리케이션들에서 모바일 데이터베이스에서 전송해야 되는 데이터 량이 급증하는 등 문제가 발생하고 있으며, 이런 환경에서는 기존 기법들의 사용이 적합하지 않아 어플리케이션의 서버 대기 시간이 길어지는 등 서비스 품질 저하로 연결된다. 본 논문에서는 데이터 캐쉬의 사용률을 높이고 통신횟수를 줄이고 무선 통신망에서의 데이터 전송량을 줄이고 모바일 데이터베이스 시스템의 성능을 향상시키기 위하여 수요에 따른 교체 우선순위 기반 모바일 동기화 정책을 제안한다. 또한 성능평가를 통하여 제안기법이 데이터 전송량을 줄이고 데이터 전송 성공 확률을 향상시켜 모바일 클라이언트가 대량의 데이터의 전송을 요청할 때 데이터 전송 효율이 향상됨을 보인다.

▸ 키워드 : 모바일 데이터 캐싱, HDC, 동기화, 우선순위

I. INTRODUCTION

Mobile database system[1,2](MDS) is a kind of distributed database system which can be used in the mobile computing environment. It is an extension of traditional distributed database. Generally, mobile database has two meanings[3,4]. One is moving person can access database server or its copy. The other is that person can move along with the copy of database. MDS can satisfy data request at any time and any place.

In mobile network environment, MDS can be divided into three layers[5,6,7] with each layer connect another by different nodes. (1) Server layer. It is fixed host node and always used as database server and synchronization server. (2) Mobile support station(MSS) or base station. It connects a number of mobile clients through wireless network. And different mobile clients connected to the same MSS are called a wireless network unit. (3) Mobile client(MC) or mobile unit(MU). It has limited storage and ability of data processing. In mobile network environment, connections of MSS and MCs are mobile, intermittent, low broadband, high delay and unstable. Therefore, traditional distributed database technology could not work efficiently in the mobile computing environment. Figure 1 describes a general architecture of mobile database system[6].

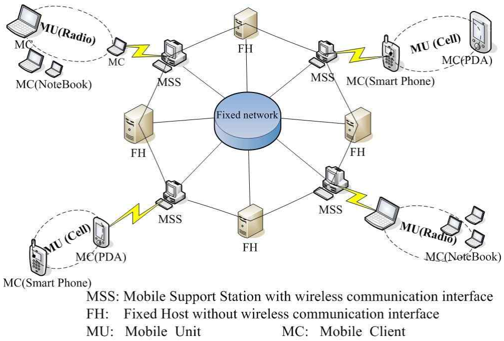

Figure 1: the architecture of mobile database system

Compared with traditional distributed database, mobile database has distinct characteristics [8,9,10,11]: mobility, intermittently, bandwidth diversity, weak reliability, asymmetric network communication, limited power, etc.

How to improve system performance and support mobile database efficient synchronous management under limited bandwidth and low channel rate is a crucial problem. Previous researches output that mobile data caching [12,13,14,15,16,17] is an effective method. It reduces frequency of accessing the database server and improve accessing speed. In mobile data caching, some data is stored in the mobile client instead of only in database server. However, the method will lead to data inconsistency. In the disconnected state, mobile data caching allows data read and write operation directly in mobile client which will lead to inconsistency between server and client. In the connected state, consistency can be guaranteed by data synchronization [2,3]. In the synchronization, when data in-demand cannot be found in the cache, mobile client should request data from the server. In this paper, we propose a new mobile data caching synchronization strategy based on in-demand and replacement priority in the mobile network environment. It can reduce quantity of wireless tra nsmission and improve transaction success rate, especially when mobile client request a large amount of data. Figure 2 shows our strategy's general flow.

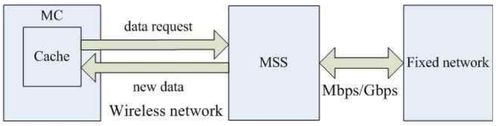

Figure 2: the flow of mobile data caching synchronization strategy

This paper is organized as follows: Section 1 introduces the concept of mobile database system and its architecture and Section 2 reviews related work on the mobile database cache. We describe in-demand and replacement priority strategy based on HDC in Section 3. In Section 4, an example and experiment of strategy are given. Finally, we conclude our work and put forward future research direction in Section 5.

II. RELATED WORK

Recently, many researchers focused on mobile data synchronization, which is a key technology in mobile database. Up to now, many approaches have been proposed on how to improve the synchronization performance of mobile database.

Xiao Zhang and Xiaofeng Meng proposed a synchronization recovery method with queue mech anism in order to reduce communication costs [2]. Wai GenYee and Ophir Frieder raised a phenomenon that if you design the update files appropriately, it'll result in significant performance improvements [17]. Niraj Tolia etc reported a system named Cedar that work in the low-bandwidth networks and get good performance at the same time. Cedar uses content addressable storage to discover the similarity between server and client results, which make that it is an easy way to use the optimistic method to solve database replica control problem [18].

Miseon Choi etc proposed a new notion of hoardset fragment ability and Split Synchronizing Mobile Transaction (SSMT) model for ubiquitous mobile client which not only assures transaction atomicity in synchronization but also reduces sync hronization cost [19,20]. [21] introduced a new algorithm called Hot Data Caching (HDC). It is able to satisfy the data needs of maximum number of transactions running at MUs. This can be achieved if the desired data sets for the present and also for recent future sets of transactions are identified reasonably accurately and made them available locally.

Mi-Young Choi etc proposes an Synchronization Algorithms Based on Message Digest (SAMD) algor ithm based on message digest in order to facilitate data synchronization between server-side database and mobile database [22]. The SAMD uses only the standard SQL functions for the synchron ization. Therefore, the SAMD algorithm can be used in any combinations of server-side database and mobile. This feature is important in order to build efficient mobile business systems because the upcoming mobile business environment has heter ogeneous characteristics in which diverse mobile devices, mobile databases, and RDBMS exist.

The HDC algorithm [21] is able to reduce transm ission of quantity which is a key technology in the mobile database. But, there are two disadva ntages. One is its data request sequence which doesn't take consider the invalid data item. The other is its replacement strategy which uses the queue (First in first out) cannot retain the frequent data. To solve these problems, our paper suggests two parameters: in-demand priority and replacement priority. They are used in the request sequence and replacement sequence.

III. MOIBLE DATA CACHING SYNCHRONIZATION STRATEGY

3.1 Definitions

In the mobile database, the unit of data operation is transaction. Additionally, one transaction always involves one or more data items. It is hoped to complete the most transactions with less data items request. These data items should be called Hot Data.

Since mobile database has to maintain weak consistency, we set a tolerable update time called threshold ⊿T(value of threshold should be decided based on the concrete system. The higher real-time requests, the smaller value of threshold is.), that is to say, during the threshold we can assume data items valid. Otherwise, the current data items are "dirty". It means current data items are invalid.

Hot data [19] set is obtained by establishing the transaction-data (T-D) matrix and reducing.

Assume all transactions which need request new data items T={Ti| Ti is a transaction, i=1, 2, …, n},n is the number of transaction.

We use D to represent all the data items involved in T. D={Di| Di is a data item, i=1, 2, …, m}, m is the number of data item.

3.2 Cache request strategy

The paper uses cache request strategy whose request queue order is hot data at first, and then data items ordered by in-demand priority.

High in-demand priority should include following respects: high weight, high access probability, low complexity of data, update time interval length and so on.

(1) Weight. It can be considered as a degree of concerned or real-time demand. In a table, data of one column represents same attribute, so the weights are same too. For each column, we give a weight in advance. In the request queue of data, the column with high weight is preferred.

(2) Probability of access. It indicates times of user access. The more times the data item accessed, the prior we should request. We use p to represent it.

(3) Complexity of data. It is hoped that data complexity O(d) is small to save wireless network channel resources.

(4) Update time interval. It refers to the difference between current time and data item's update time in cache. After hot data set established, there will be unceasingly new transaction to join the transaction queue. When it achieves certain extent, new T-D matrix will be established, reduced and we get new hot data. During synchronization, some data may not be hot data or have low weight, low visit probability and high data complexity. At the moment, these data may not be updated for long time and cause a phenomenon-starve to death. With update time interval, we can avoid the problem.

So, the in-demand priority F is calculated as follows:

$$F = \frac{w_i p \Delta t}{O(d)} (0 < w_i < 1, 0 < p < 1, 0 < \Delta t < T)$$

T is data invalid time. P is counted by mobile client and gets along with new transactions. It is created by trigger.

3.3 Cache replacement strategy

We define an array to store replacement priority recorded as P. For each data item, P is calculated as follows:

$$P = k * \text{Hit times} / (\text{Tnow} - \text{Tupdate time})$$

k is a factor defined by user. It means relative importance of hit time and update interval time.

The paper uses cache replacement strategy which combines hit time and update interval time to manage cache. That is, if the MU cache is full, it'll replace the data item whose P value is small.

3.4 Algorithm description

We describe the proposed algorithm in Pseudo as follows:

```
Algorithm: mobile data caching synchronization strategy
based on in-demand and replacement priority
Require:
Preprocess transaction Array T[number of transaction] and
get the data item array D[number of data items];
array H[]; // H is used to store Hit data items
array M[]; //M is used to store Missed data items
array O[]; // O is used to store hot data items
array C[]; // C is used to store data items which are not
hot data.
Step1: Initialize the H and M. If D[i] is in cache and valid,
then put D[i] into H, else put it into M.
Step2: Establish the T-D matrix and get the hot data set.
Step3: Initialize O and C. if D[i] is hot data, then put D[i]
into O, else put it into C.
Step4: Output request queue Q. If M[i] is in O, put M[i]
into Q, else compute M[i]' in-demand priority F. According
to F's order, put M[q] to Q.
Step5: Output replacement priority R. If D[i] is in H, add 1
to D[i]'s hit time, then compute replacement priority P of
D[i]. If cache is full, replace  D[i] with minimum
replacement priority.
```

In order to illustrate the algorithm clearly, we explain working of data identification phase of algorithm with a simple example. As Table 1 and Table 2 show, we assume size of MUs workload (num of transactions that can be scheduled concurrently) is 7. D1, D2… represent data items and T1, T2... represent transactions. D3, D8, D10 (replacement priority is 1) are valid and in the cache, D6 (replacement priority is 1) is invalid but in the cache. As the table shows, a "1" in the matrix indicates that the corresponding transaction needs that data item. All 1's in each column is added. A lowest value column sum identifies the data item, which is accessed by least number of transactions.

Table 1: T-D matrix (first 7 transactions)

data item / transaction	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
T1	1		1							
T2			1				1			
T3	1							1		1
T4					1		1			
T5			1			1				
T6	1						1			
T7			1	1		1	1	1		1
SUM	3		4	1	1	2	4	2		2

Table 2: T-D matrix (latter 7 transactions)

T8	1									1
T9		1		1						
T10									1	
T11	1		1				1			
T12			1							
T13	1		1					1		
T14			1			1		1		1
SUM	3	1	4	1		1	1	2	1	2

At the first time, we handle first 7 transactions (Table 1). The lowest value of column data and the corresponding transactions are removed from the matrix. Thus, D4, D5, D6, D8, D9, D10 and their corresponding transactions are removed, which is shown in table 3.

Table 3: after reduction of the first T-D matrix

data item / transaction	D1	D3	D7
T1	1	1	
T2		1	1
T6	1		1
SUM	2	2	2

So, hot data items are D1, D3, and D7.

Data items which need request are (D1, D7), (D4 D5, D6). And D1, D7 are hot data, so we apply for them at first. Thus, we only need 2 data items and complete 3 transactions, that is to say, 20% data items complete 43% transactions. It does a lot to improve success rate of transaction. Meanwhile, hit time of D3, D6, D8 and D10 is 1. Hit time of D1, D7, D4 and D5 is 0(Initialized hit time of new data item is 0).

Before rest transactions arrive, data items in cache are D1, D3, D4, D5, D6, D7, D8, D10. We assume D3 is invalid. We process latter 7 transactions from Table 2. After reduction, the result lists in table 4.

Table 4: after reduction of the second T-D matrix

data item / transaction	D1	D3	D8
T12		1	
T13	1	1	1
SUM	1	2	1

After reduction, hot data is D1, D3 and D8.

Because D3 is invalid, data items that need apply are (D3) , D2, D9. D3 is hot data, so we apply for them at first. Thus we only need 1 data item and complete 2 transactions. Meanwhile, hit time of D3, D6, D8 and D10 is 2. Hit time of D1, D4,and D7 is 1. Hit time of D2 and D9 is 0.

When new data item arrives at mobile client, we just use replacement priority described in previous part of paper to determine which the unit to replace if the cache is full.

3.5 Algorithm analysis

Proposed algorithm use HDC to assure the data item which is accessed most time by users update fastest. Hence, it is have better real-time and user-oriented.

Compared with hot data caching, if we found data items which are not in cache and not current hot data, we put it into the request queue according to F (in-demand priority, from large to small order). The proposed algorithm can ensure the validity of data item. It can prevent some data items in cache, which is invalid, to be read as "dirty data". In this way, our algorithm can raise the transaction success rate. For instance, if D3 in the example is read by T12, T13, it is inevitable to roll back for T12, T13. In addition, the replacement strategy also can effectively avoid the frequently use data to be replaced when cache is full. Thus, it is helpful to improve data utilization and reduce the wireless transmission quantity.

IV. PERFORMANCE EVALUATION

4.1 Experiment Environment

Our experiment use 5 MCs whose CPU frequency is 1GHz~1.2GHz and a server whose CPU is Intel(R) Core, 2GHz as MSS to implement the mobile database system. In addition, the wire network bandwidth is 2Mbps. The proposed algorithm was implemented using JAVA and were linked with databases using JDBC (Java Database Connectivity). Database server is implemented on SQL Server. And Database size is 5000 entries, and transaction size is between 2 to 10 data items. There are 1000 transactions to perform read operation and 2000 transactions to performance write operation. We use two parameters showed in Table 5 to analysis our algorithm's performance.

Table 5: performance evaluation parameters

Parameter	Description
Transmission quantity	the wireless data transmission quantity between MCs and MSS.
Transaction success rate	avg. number of successful transaction / (number of successful transactions + number of roll back transactions) of all MCs.

4.2 Performance Evaluation

Assume if there is no data item which is required in transaction in mobile cache, we apply it for server directly. We get below table from the experiment.

Figure 3 shows the performance of proposed algorithm using different data items.

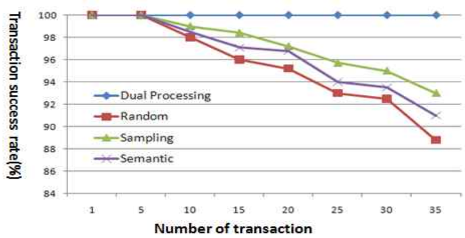

Figure 3: the average transaction success rate of MCs

Figure 4 implies that when the transaction increases , proposed algorithm's transaction success rate is much higher than HDC. Our algorithm takes validity of data item into consideration. If data item is invalid, we request it from server. In this way, we ensure the validity of data items, reduce the possibility of reading "dirty" data items and improve the transaction success rate.

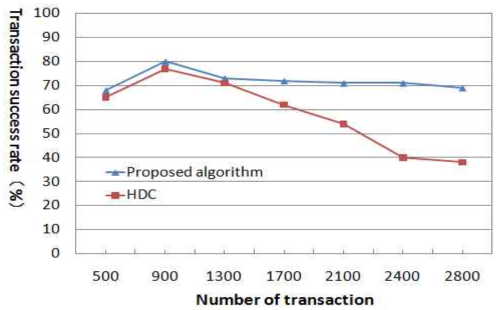

Figure 4: the average transaction success rate of MCs

And from figure 5, we can see our algorithm's can reduce wireless transmission quantity efficiently. When number of transaction increased rapidly, cache must be full and the replacement algorithm is more important. our algorithm can make the replacement queue according value of replacement priority. In this way, data item with higher hit possibility will not be replaced, improve the data items' utilization rate and reduce transmission quantity.

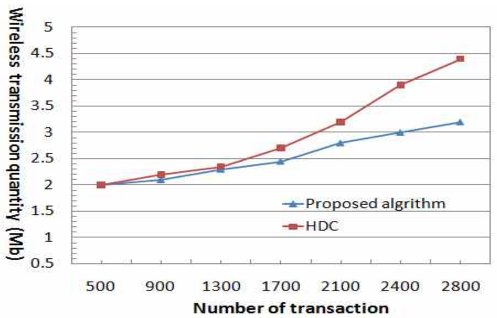

Figure 5: the wireless transmission quantity of a MSS

So we can get our algorithm works a lot in large number of transaction processing and reduces quantity of wireless transmission.

V. CONCLUSIONS

With development of mobile communication, mobile database application is widely used in the military, commercial, data transmission of mine environmental monitoring, etc. However, mobile database is primarily in academic phase, and has not the unified international standards, and lacks universal commercial mobile database product. In this paper, we proposed a mobile data caching synchronization strategy based on in-demand and replacement priority to have better real-time, user-oriented and improve data utilization. We illustrate it by example and experiment which shows that it can do well in reducing the wireless transaction quantity and dealing with MC transactions efficiently in large number of data environment.

REFERENCE

[1] Sang-ouk Kim, Se-Bong Oh, Sung-Young Son, Jin-Ho Lee, "The framework for synchronization in embedded database environment", Journal of Computing Science and Engineering, Vol.20, Num.7, pp.14-21, 2002.

[2] Zhang Xiao,Meng Xiaofeng,Wang Shan,"Transactional synchronization based on data synchronization object in Lite/2", Journal of Software, May. 2002, pp.937-945.

[3] A.Ranganathan, J.Al-Muhtadi, S.Chetan,"MiddleWhere: A Middleware for Location Awareness in Ubiquitous Computing Applications", Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware, 2004.

[4] Hyuk Lee, Yeon Lee, weon-IL Chung, Hae-Young Bae, "Spatial-Sensor Observation Service for Spatial Operation of GeoSensor", Journal of The Korea Society of Computer and Information, VOL.16, NO.11, November 2011.

[5] Dua,R., Bhandari,s., "Recovery in mobile database

system", International Conference on Wireless and Mobile Communications (ICWMC'06), July 2006, P32-32.

[6] D.Walbom, Panos K., "Supporting Semantics-Based Transaction Processing in Mobile Database Applications", Proceedings of the IEEE Symposium on Reliable Distributed, 1995.

[7] Wang ShunYan, Zhong Luo, Cao YongLiang. A Data Synchronization Mechanism for Cache on Mobile Client. IEEE, 2006.

[8] Yang Li, Xuejie Zhang, Yun Gao, "Object-Oriented Data Synchronization for Mobile Database over Mobile Ad-hoc Networks". Information Science and Engineering, 2008.

[9] Chen Shu, Jiang Ningkang, "Synchronization mechan ism of mobile database based on multi-aspects integrity detection", Journal of Computer Applic ations, 2009.

[10] Z. Dzolkhifli, H.Ibrahim,L.S. Affendey, "A Framework for Caching Relevant Data Items for Checking Integrity Constraints of Mobile Database", IEEE, 2008.

[11] Zejun Wu, Yan Li, Hae-Young Bae, "The Design and Implementation of Real-time Data Stream Server for Continuity of Care Record", Journal of The Korea Society of Computer and Information, VOL.16, NO.12, December 2011.

[12] Zhang Jian, Shuai bing, Liang xiang-jun, "Research on Synchronization Model Based on Mobile Data base Replication Technology", Modern Comput er, 2008.

[13] Gye-Jeong Kim, Seung-Cheon Baek, Hyun-Sook Lee, Han-Deok Lee, etc, "LGeDBMS: a small D BMS for embedded system with flash memory", 32nd international conference on very large data bases, pp. 1255~1258, 2006.

[14] Teixeim Neto M C, SMgado A C, "Hoarding and prefetching for mobile databases", 5th IEEE/ACIS International Conference on Computer and Information Science, July 2006, P219-224.

[15] Kumar V, "Mobile database systems", John Wiley& Sons, Inc, 2006.

[16] Yazhou Jiao, Zhigang Jin, Zhuoqun Ma, "A Cros s-layer Method to Improve Mobile Database Synchronization Performance". IEEE, 2009.

[17] Wai Gen Yee, Ophir Frieder, "Scalable synchroniza -tion of intermittently connected database clients," In Proceedings of sixth International Conference on Mobile Data Management(MDM'05), May 2005, pp.299-303.

[18] Niraj Tolia, M.Satyanarayanan, Adam Wolbach, "Improving mobile database access over wide-area networks without degrading consistency", In proceedings of the 5th International Conference on Mobile Systems, Applications and Services(MobiSys'07), Jun 2007, pp.71-84.

[19] Miseon Choi, Wonik Park, Young-Kuk Kim, "A split synchronizing mobile transaction model," In proceedings of the 2nd international conference on Ubiquitous information management and communicationin, Suwon, Korea, Feb. 2008, pp.196-201.

[20] M.Choi, W.Park, Y.oung-KuK Kim. "A Split Synch ronizing Mobile Transaction Model", Proceedings of the 2nd international conference on Ubiquitous information management and communication,2008.

[21] Kumar V, Prabhu N, Chrysanthis P K. "HDC-hot data caching in In mobile database systems", The 3rd ACS/IEEE International Conference on Com puter Systems and Applications, 2005: 37.

[22] Mi-Young Choi, Eun-Ae Cho, Dae-Ha Park,C hang-Joo Moon, Doo-Kwon BaikA, "Database Synchronization Algorithm for Mobile Devices". IEEE. 2010.

저 자 소 개

Jihua Zhao

2005~2009 : Chongqing University of Posts and Telecommunications. (B.S)

2009~now : Chongqing University of Posts and Telecom munications. (M.S)

Research Interests : Mobile Database, Database Mining, Sensor Netwo rk

Email : zhaojinhua_anna@hotmail.com

Xia Ying

1986~1989 : Electronic Science Department, ChongQing Univ. of Post and Telecom., PR China(B.S)

1999~2001 : Computer Information Engineering Depart. Inha Univ, (M.S)

2006~present : Computer application department, Sou th-west JiaoTong Univer sity, Ph.D course

2003~now : Prof. of Computer department of ChongQing Univ. of Post and Telecom., PR China

2006년~now : ChongQing Univ. of Post and Telecom., PR China, Director of Sino-Korea GIS Research Center

Research Interests : Database, GIS, LSB Policy and Technology, Data Stream

Email : xiaying@cqupt.edu.cn

Soon-Jo Lee

1985 : Electronic Computing Depart. Inha Univ (B.S)

1987 : Electronic Computing Department, Inha Univ (M.S)

1995 : Electronic Computing Department, Inha Univ (Ph.d)

1995~1997 : Electronic Com puter Department, Prof.

1997~present : Dept. of Computer Engineering, Seowon Univ, Prof.

Research Interests: Database, Spatial database.

Email: sjlee@seowon.ac.kr

Hae-Young Bae

1974 : Physical Engineering Depart. Inha Univ (B.S)

1978 : Computer Science and Engineering Depart. Yonsei Univ (M.S)

1989 : Computer Science and Engineering Depart. Soongsil Univ (Ph.d)

2006~2008: Director of Inha University Graduate School

1982~present : Computer Information Engineering Depart. Inha Univ, Prof.

Research Interests: Database, Spatial database.

Email : hybae@inha.ac.kr