Shared Data Allocation in a Mobile Computing System: Exploring Local and Global Optimization

Wen-Chih Peng and Ming-Syan Chen, *Fellow*, *IEEE*

Abstract—In this paper, we devise data allocation algorithms that can utilize the knowledge of user moving patterns for proper allocation of shared data in a mobile computing system. By employing the data allocation algorithms devised, the occurrences of costly remote accesses can be minimized and the performance of a mobile computing system is thus improved. The data allocation algorithms for shared data, which are able to achieve *local optimization* and *global optimization*, are developed. Local optimization refers to the optimization that the likelihood of local data access by an *individual* mobile user is maximized whereas global optimization refers to the optimization that the likelihood of local data access by *all* mobile users is maximized. Specifically, by exploring the features of local optimization and global optimization, we devise algorithm SD-local and algorithm SD-global to achieve local optimization and global optimization, respectively. In general, the mobile users are divided into two types, namely, frequently moving users and infrequently moving users. A measurement, called *closeness measure* which corresponds to the amount of the intersection between the set of frequently moving user patterns and that of infrequently moving user patterns, is derived to assess the quality of solutions provided by SD-local and SD-global. Performance of these data allocation algorithms is comparatively analyzed. From the analysis of SD-local and SD-global, it is shown that SD-local favors infrequently moving users whereas SD-global is good for frequently moving users. The simulation results show that the knowledge obtained from the user moving patterns is very important in devising effective data allocation algorithms which can lead to prominent performance improvement in a mobile computing system.

Index Terms—User moving patterns, mobile computing, shared data allocation, mobile databases.

1 Introduction

Due to recent technology advances, an increasing number of users are accessing various information systems via wireless communication. Such information systems as stock trading, banking, and wireless conferencing, are being provided by information services and application providers [14], [16], [28], and mobile users are able to access such information via wireless communication from anywhere at any time [2], [8], [23], [26].

For cost-performance reasons, a mobile computing system is usually of a distributed server architecture [1], [16], in which a service area, referring to the converge area where the server can provide services to mobile users, contains one or many cells where a cell refers to a communication area covered by a base station. In general, mobile users tend to submit transactions to servers nearby for execution so as to minimize the communication overhead incurred [14], [16]. Data objects are assumed to be stored at servers to facilitate coherency control and also for memory saving at mobile units [25], [27]. Since the architecture of a mobile computing system is distributed in nature, data replication is helpful because it is able to improve the execution performance of servers and facilitate the location lookup of mobile users [13], [25], [27]. The replication scheme of a data object involves how many replicas of that object to be created, and to which servers those replicas are allocated. Clearly, though avoiding many costly remote accesses, the approach of data replication increases the cost of data storage and update. Thus, it has been recognized as an important issue to strike a compromise between access efficiency and storage cost when a data allocation scheme is devised.

Various wireless data networking technologies, including UMTS [22], IMT 2000 [3], and W-CDMA [7], [15] [18], have been developed recently. The wireless applications are expected to become even more popular as the technology of the third generation mobile network (3G) advances and the popularity of portable devices increases. For example, Advanced Traffic Information Systems (ATIS) [1], [12], [24] are systems developed to provide useful information for travelling and driving. With portable computers, drivers are able to periodically obtain updated information, via the wireless communication, on traffic reports, street maps, and entertaining information, to name a few. From the perspective on quality of services, data access rates of drivers are fixed. These traffic reports are examples of shared data (referring to those data to be accessed by a group of users) considered in this paper.

It is noted that various data allocation schemes have been extensively studied in the literature [21], [25], [27]. However, the data allocation schemes for traditional distributed databases are mostly designed in static manners, and the user moving patterns, which are particularly relevant to a mobile computing system where users travel between service areas frequently, were not fully explored. Note that the server is expected to take over the transactions

• The authors are with the Department of Electrical Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei, Taiwan, Republic of China. E-mail: wcpeng@csie.nctu.edu.tw, mschen@cc.ee.ntu.edu.tw.

Manuscript received 20 May 2002; revised 25 May 2004; accepted 17 Aug. 2004; published online 23 Feb. 2005.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 116583.

1045-9219/05/$20.00 © 2005 IEEE Published by the IEEE Computer Society

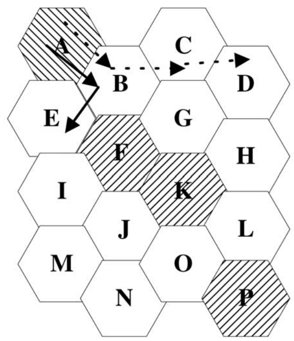

Fig. 1. An example scenario of shared data allocation problem.

submitted by mobile users and static data allocation schemes may suffer severe performance problems in a mobile computing system. Without loss of generality, an example network topology of a mobile computing system is given in Fig. 1. As shown in Fig. 1, there are 16 servers and the number of replicated server is 4. Suppose that the shared data are replicated statically at sites A, F, K, and P under the data allocation schemes for traditional distributed databases. Assume that the mobile user U_1 is found to frequently travel in service areas of A, B and E (i.e., $\{A, B, E\}$ is called the moving pattern of mobile user U_1) and the mobile user U_2 frequently moves in the service areas of A, B, C, and D. As can be seen in Fig. 1, the solid line is an example moving path of the mobile user U_1 and the dotted line is that of the mobile user U_2. It can be seen that the advantage of having replicas on F, K, and P cannot be fully taken by mobile users U_1 and U_2, and the extra cost of maintaining those replicas is not justified by the moving patterns of users U_1 and U_2. Note that similarly to the calling patterns of users, it is envisioned that most users tend to have their own user moving patterns since the behaviors of users are likely to be regular [5], [25], [29]. In order to improve the system performance, efficient data allocation schemes based on moving patterns of mobile users are very important in a mobile computing environment. Since each mobile user has his/her own moving pattern, how to select proper sites for shared data allocation is the very problem we shall address in this paper.

In this paper, we devise data allocation algorithms that can utilize the knowledge of user moving patterns for proper allocation of shared data. By employing the shared data allocation schemes devised, the occurrences of costly remote accesses can be minimized and the performance of a mobile computing system is thus improved. Explicitly, the data allocation schemes for shared data, which are able to achieve *local optimization* and *global optimization*, are developed. Local optimization refers to the optimization that the likelihood of local data access by an *individual* mobile user is maximized whereas global optimization refers to the optimization that the likelihood of local data access by *all* mobile users is maximized. By exploring the features of local optimization and global optimization, we devise algorithm SD-local and algorithm SD-global to achieve local optimization and global optimization, respectively. In this paper, it is assumed that the mobile users who have regular moving behaviors can be further divided into two types, namely, frequently moving users and infrequently moving users. Such an assumption is justified by the calling data records provided by a cellular phone company in Taiwan. It is expected that this phenomenon also exists in a mobile computing environment. A measurement, called *closeness measure* which corresponds to the amount of the intersection between the set of frequently moving user patterns and that of infrequently moving user patterns, is derived to assess the quality of solutions resulted by SD-local and SD-global. Performance of these data allocation schemes is analyzed and sensitivity analysis on several design parameters is conducted. From the analysis of SD-local and SD-global, it is shown that SD-local favors infrequently moving users and SD-global is good for frequently moving users. Our simulation results show that the knowledge obtained from the user moving patterns is very important in devising effective shared data allocation schemes which can lead to prominent performance improvement in a mobile computing system.

A significant amount of research efforts has been elaborated upon issues of data allocation in distributed systems [9], [19], [21], [25], [27]. We mention in passing that the authors of [27] proposed a data distribution scheme that is based on the read/write patterns of the data objects. Given some user calling patterns, the authors of [25] proposed an algorithm that employed the concept of minimum-cost maximum-flow to compute the set of sites where user profiles should be replicated. Without fully exploiting user moving patterns, the attention of the study in [25] was mainly paid to the distribution of location data for mobile users. Schemes for personal data allocation were explored in [20], whose attention was mainly paid to developing mining procedure to cope with personal data allocation, but not for the shared data allocation explored in this paper. With its criterion being intrinsically different from that of personal data allocation, the shared data allocation, in our view, is particularly useful for many services and applications in a mobile computing environment.

The contributions of this paper are twofold. We not only devise data allocation for shared data in a mobile computing system, but also in light of user moving patterns obtained, optimize the data allocation schemes devised. To the best of our knowledge, none of the prior work has fully utilized user moving patterns for shared data allocation, let alone devising schemes to achieve the local optimization and global optimization and conducting the corresponding performance analysis. These features distinguish this paper from others. With the rapid advances in wireless technologies, the mobile computing systems are becoming widely available nowadays. The fast increase in mobile applications justifies the timeliness and importance of this study.

This paper is organized as follows: Preliminaries are given in Section 2. Shared data allocation algorithms based on user moving patterns are developed in Section 3. Experimental results are presented and analyzed in Section 4. This paper concludes with Section 5.

2 Problem Formulation

Note that the number of servers and the number of replicated servers are fixed due to that the cost of maintaining a larger number of servers is prominent. In this paper, we devise data allocation algorithms that can utilize user moving patterns to determine the set of replicated servers for proper shared data allocation. Since each mobile user has his/her own moving pattern, how to select proper sites for shared data allocation is the very problem we shall deal with in this paper. The problem that we study in this paper can be formally defined as follows:

TABLE 1
Description of Symbols

Description	Symbol
The frequent set of mobile user U_i	FS_i
The probability of local access hit for mobile user U_i	L(U_i)
The set of total servers in a mobile computing system	S
The set of replicated servers	R
Number of moving paths for mobile user U_i	n_i
Number of mobile users	N
The threshold value to identify frequent moving mobile users	P_T

Problem of shared data allocation based on user moving patterns: Given the number of mobile users with their moving patterns, the number of servers and the number of replicated servers, we shall determine the proper set of sites to which shared data are allocated with the purpose of maximizing the number of local access of shared data. In this paper, with the proper allocation of shared data, the number of local access of shared data is improved and the properties of data objects are read-only in order to fully focus our problem on design the shared data allocation based on user moving patterns. Table 1 shows the description of symbols used in modeling the problem. Fig. 2 shows the problem formulation of allocating shared data where the number of mobile users is 3. The set of total servers is expressed by S, where |S| is the total number of servers. Denote the set of replicated sites for shared data as R. The union set of moving patterns for mobile user U_i is expressed by FS_i (standing for frequent set), where $|FS_i|$ is the number of distinct sites within the set FS_i. The number of moving paths for mobile user U_i is denoted by n_i, where a moving path is a sequence of servers accessed by a mobile user. Clearly, the probability of local access of mobile user U_i, denoted by $L(U_i)$, is proportional to $\frac{|R \cap FS_i|}{|FS_i|}$, which is formulated as follows:

$$L(U_i) = f * \frac{|R \cap FS_i|}{|FS_i|}, \text{where } f \text{ is a hit coefficient and } 0 < f < 1. \quad (1)$$

Consider the mobile user U_1 in Table 2, where the network topology is shown in Fig. 1. Assume that without exploring user moving patterns, the set of replicated sites $R = \{A, F, K, P\}$ and the value of f is 0.8. From Table 2, the set of FS_1 can be obtained by unifying two moving patterns of mobile user U_1 into one set, i.e., $FS_1 = \{AE\} \bigcup \{ABC\} = \{ABCE\}$. It can be verified that the set of $R \cap FS_1$ is $\{A\}$. Then, we have the estimated probability of local access of mobile user U_1 is $0.8 * \frac{1}{4} = 0.2$. Since each mobile user has his/her own moving patterns, how to select proper sites for shared data allocation, i.e., R, is a very important issue which will be dealt with in this paper.

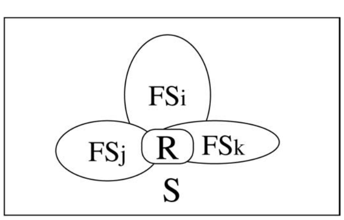

Fig. 2. The problem of shared data allocation where the number of mobile users is 3.

As mentioned before, according to the number of moving paths of mobile users, mobile users can be divided into two types, namely, frequently moving users and infrequently moving users. To facilitate the presentation, we denote P_T as the threshold to determine whether the mobile user belongs to the group of frequently moving users or not.

Definition 1. *The union set of frequent sets of frequently moving users is defined as* $F_{FS} = \bigcup_{\forall i, 1 \le i \le N \text{ and } n_i \ge P_T} FS_i$.

Definition 2. *The union set of frequent sets of infrequently moving users is defined as* $U_{FS} = \bigcup_{\forall i, 1 \le i \le N \text{ and } n_i < P_T} FS_i$.

To quantify how closely F_{FS} approximates U_{FS}, we use a closeness measure, denoted by $C(F_{FS}, U_{FS})$, that returns normalized value in $[0, 1]$ to indicate the closeness between F_{FS} and U_{FS}. The larger the value of $C(F_{FS}, U_{FS})$ is, the more closely F_{FS} approximates to U_{FS}. $C(F_{FS}, U_{FS})$ is formulated as follows:

$$C(F_{FS}, U_{FS}) = \frac{|F_{FS} \cap U_{FS}|}{|F_{FS} \cup U_{FS}|}.$$

For the example profile in Table 2, assuming the value of P_T is 500. U_1 is the frequently moving user (with $n_1 = 1,500$ movements), and U_2, U_3, and U_4 are infrequently moving users (with $n_2 = 350$, $n_3 = 300$, and $n_4 = 200$ movements, respectively). Also, the set of F_{FS} is $\{ABCE\}$ (i.e., FS_1) and the set of U_{FS} is $\{BCDFGK\}$ (i.e., $FS_2 \bigcup FS_3 \bigcup FS_4$). It can be verified that the value of $C(F_{FS}, U_{FS})$ is 0.25 (i.e., $\frac{2}{8}$). As can be seen later, the closeness measure between F_{FS} and U_{FS} influences the solution quality resulted by shared data allocation algorithms.

3 Shared Data Allocation Algorithms Based on Moving Patterns

In Section 3.1, we develop two shared data allocation algorithms, SD-local and SD-global, to improve the performance of a mobile computing system. An analysis of algorithms SD-local and SD-global is given in Section 3.2.

TABLE 2
An Example Profile for Illustrating Shared Data Allocation Schemes

User i	Moving Patterns	Frequent Set FS_i	Number of moving paths n_i
U_1	AE, ABC	ABCE	1500
U_2	BC, GK	BCGK	350
U_3	BCD	BCD	300
U_4	CGK	CGK	200

3.1 Data Allocation for Shared Data

First, we describe the data allocation scheme in a fixed pattern. In light of the user moving patterns determined, we develop two data allocation algorithms based on moving patterns for shared data allocation (to be referred to as algorithm SD-local and algorithm SD-global).

3.1.1 Data Allocation Scheme in a Fixed Pattern

In the scheme which allocates data in a fixed pattern (referred to as DF), the replication sites are determined when the database is created. Explicitly, the number of replicated sites and the sites at which the shared data can be replicated are predetermined. Though being adopted in some traditional distributed database systems due to its ease of implementation [19], [27], DF is not suitable for mobile computing environments where mobile users move frequently. In our experimental studies in Section 4, DF will be implemented for comparison purposes. As can be seen later, DF suffers from poor performance since it does not take user moving patterns into consideration.

3.1.2 Shared Data Allocation Based on Moving Patterns

As described before, shared data refers to those data that are used by many mobile users. Example shared data include public information, cooperative information, etc. By properly determining the set of replicated servers used by a group of mobile users, data allocation for shared data is able to increase the local data access ratio in the sense of both local and global optimization. Local optimization refers to the optimization that the likelihood of local data access by an *individual* mobile user is maximized, meaning that the probability of average local access is maximized. Accordingly, we have the following objective function for local optimization,

$$OPT_{local}(N) = \frac{1}{N}\sum_{i=1}^{N} L(U_i) = \frac{1}{N}\sum_{i=1}^{N} f * \frac{|R \cap FS_i|}{|FS_i|}$$
$$= \frac{f}{N}\sum_{i=1}^{N}\frac{|R \cap FS_i|}{|FS_i|},$$

where N is the number of mobile users and f is the hit coefficient. In contrast, global optimization refers to the optimization that the likelihood of local data access by *all* mobile users is maximized, meaning that the number of total local accesses is maximized. Hence, the objective function for global optimization can be formulated as follows:

$$OPT_{global}(N) = \sum_{i=1}^{N} L(U_i) * n_i = \sum_{i=1}^{N} f * \frac{|R \cap FS_i|}{|FS_i|} * n_i$$
$$= f\sum_{i=1}^{N}\frac{|R \cap FS_i|}{|FS_i|} * n_i,$$

where N is the number of mobile users, f is the hit coefficient, and n_i is the number of moving paths for mobile user U_i.

With the user moving patterns obtained, we can develop shared data allocation algorithms to determine the set of replicated servers. The shared data allocation algorithms proposed are greedy in nature and their performance will be evaluated in Section 4 experimentally. Note that moving patterns of mobile users may contain different large k-moving sequences, denoted as L_k, where a k moving sequence is called a large k-moving sequence if there are a sufficient number of moving paths containing this k-moving sequence [20]. A large moving sequence can be determined from all moving paths of each individual user based on its occurrences in those moving paths. We first convert these L_k's into L_2's and the allocation of shared data will be made in accordance with the occurrences of these L_2's. By exploiting the objective function of local optimization, we develop algorithm SD-local. In order to maximize the objective function of local optimization, the set of R should include as many sites that are frequently found in their user moving patterns as possible so as to maximize the value of $\sum_{i=1}^{N}\frac{|R \cap FS_i|}{|FS_i|}$. Thus, in algorithm SD-local, we use the user occurrence count of L_2, where the user occurrence count of L_2 is the number of mobile users whose moving patterns contain that L_2. An example profile for the counting in algorithm SD-local is given in Table 3. For example, since {AB} can only be found in the moving patterns of U_1 the user occurrence count of {AB} is one. Also, since U_1, U_2, and U_3 contain {BC} in their moving patterns, the user occurrence count of {BC} is 3. An L_2 with a larger value of user occurrences means that this pair is frequently found in moving patterns of mobile users. Hence, as mentioned above, those L_2 pairs with larger values of user occurrences should be included in the set of R so as to maximize the objective function of local optimization.

Algorithm SD-local: /* Performing SD-local for shared data allocation */
Input: All user moving patterns of mobile users
Output: The set of replicated servers, i.e., **R**
begin

TABLE 3
An Example Profile for the Counting in SD-Local and SD-Global

L_2	User occurrence count for SD-local	Movement occurrence count for SD-global
AB	1	$n_{AB}(U_1)$= 800
BC	3	$n_{BC}(U_1)+n_{BC}(U_2)+n_{BC}(U_3)$= 400+50+150=600
CD	1	$n_{CD}(U_3)$=200
CG	2	$n_{CG}(U_2)+n_{CG}(U_4)$=250+150=400
GK	2	$n_{GK}(U_2)+n_{GK}(U_4)$=250+100=350
AE	1	$n_{AE}(U_1)$=500

1. Determine, from all user moving patterns, user occurrence counts of all frequent $L_2's$
2. **Repeat** Until $|\mathrm{V}| \leq 0$; /* V is the number of replicated servers yet to determine */
3. **begin**
4. Include those $L_2's$ that have maximal user occurrence count from all $L_2's$ into the set c-max. Also, c denotes an L_2 pair in c-max.
5. **if** $|\mathrm{R}| = 0$ /* R is the set of replicated servers */
6. **begin**
7. Choose an L_2 pair from c-max;
8. Include this L_2 pair into R;
9. $|\mathrm{V}| = |\mathrm{V}| - 2$;
10. **end**
11. **else if** $(\exists \mathrm{c} \in \mathrm{c-max} \text{ and } \mathrm{R} \cap \mathrm{c} \neq 0)$
12. **begin**
13. In c-max, choose an L_2 pair that has an intersection with pairs in R;
14. $|\mathrm{V}| = |\mathrm{V}| - 1$;
15. **end**
16. **else** /* In c-max, there is no L_2 pair that has an intersection with pairs in R */
17. **begin**
18. Choose an L_2 pair from c-max;
19. $|\mathrm{V}| = |\mathrm{V}| - 2$;
20. **end**
21. $\mathrm{R} = \mathrm{R} \cup \mathrm{c}$;
22. **end**

end

Consider the execution of SD-local as an example shown in Fig. 3, where the network topology is four by four mesh network. Let R denote the set of replicated servers identified thus far. Once the user occurrence counts of all L_2 pairs are obtained, we have the configuration shown in Fig. 3a, where the number next to each edge represents the user occurrence count of the corresponding L_2. Then, we include the L_2 which has maximal user occurrence count (i.e., $\{\mathrm{BC}\}$ according to the profile in Table 3) into the set R in line 4 of algorithm SD-local, resulting in the configuration shown in Fig. 3b. In general, if the number of replicated server, $|R|$, is not equal to the number of replicated servers required, we select, from existing L_2 pairs that have maximal user occurrence count (i.e., c-max), the one that has an intersection with pairs in R (from line 12 to line 15 of algorithm SD-local). The pair $\{\mathrm{CG}\}$ is hence selected. This step is similar to Prim's algorithm for finding minimal-cost-spanning-tree (MCST) [6]. The difference between SD-local and MCST is that even the maximal support of an L_2 pair does not have any intersection with R, this pair can still be included into R as described from line 17 to line 20 of algorithm SD-local. After the inclusion of $\{\mathrm{CG}\}$, R becomes $\{\mathrm{BCG}\}$ and Fig. 3b in turn leads to Fig. 3c. Following this procedure, we shall identify and include more proper L_2 pairs until $|R|$ reaches the number of replicated servers required (i.e., $|V| = 0$). Finally, we have the configuration in Fig. 3d and R is composed of the most frequent moving sites for all mobile users in the sense of local optimization.

On the other hand, based on the objective function of global optimization, we develop algorithm SD-global to achieve global optimization. Note that since the objective function of global optimization takes the number of moving paths into account, the movement occurrence count should be used for counting, where the movement occurrence count is the sum of all the movement occurrence counts of that L_2 from all mobile users. An illustrative example profile is given in Table 3. Let $n_{BC}(U_i)$ denote the occurrence count of $\{\mathrm{BC}\}$ in moving paths of mobile user U_i. The movement occurrence count of $\{\mathrm{BC}\}$ is thus the sum of $n_{BC}(U_1)$, $n_{BC}(U_2)$, and $n_{BC}(U_3)$. Since the moving number of mobile users is the multiplier in the objective function of global optimization, those L_2 with larger values of movement occurrences will be selected so as to maximize the value of the objective function of global optimization. In

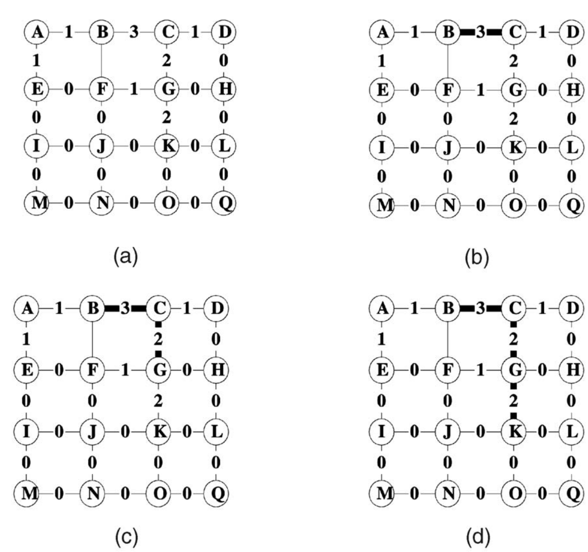

Fig. 3. An execution scenario of algorithm SD-local. (a) The original configuration, (b) include $\{\mathrm{BC}\}$ to R, (c) include $\{\mathrm{CG}\}$ to R, and (d) include $\{\mathrm{GK}\}$ to R.

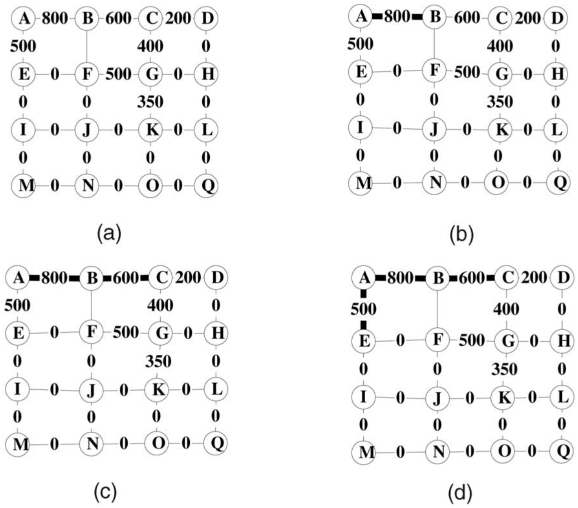

Fig. 4. The execution scenario of algorithm SD-global. (a) The original configuration, (b) include $\{AB\}$ to R, (c) include $\{BC\}$ to R, and (d) include $\{AE\}$ to R.

algorithm SD-global, the selection of R should include those L_2 pairs that have the larger user movement occurrence count so as to maximize the value of $\sum_{i=1}^{N} \frac{|R \cap FS_i|}{|FS_i|} * n_i$. The algorithmic form of SD-global is in essence the same as that SD-local except the former substitutes movement occurrence counts for user occurrence counts of the latter in the corresponding line 1. In algorithm SD-global, the movement occurrence count of L_2 is used (line $1'$) for counting. With the same profile in Table 3, Fig. 4 shows the execution scenario of SD-global, where the number next to each edge represents the movement occurrence count of the corresponding L_2. The set of replicated servers by SD-global can be obtained similarly.

Algorithm SD-global /* Performing SD-global for shared data allocation */
Input: All user moving patterns of mobile users
Output: The set of replicated servers, i.e., **R**
begin
$1'$. Determine, from the counting statistics of mining user moving patterns [20], *movement occurrence counts* of all frequent $L_2's$
/* line 2 to line 22 are the same as algorithm SD-local */
end

Table 4 shows the example execution by algorithm SD-local and algorithm SD-global with the profile given in Table 2. The local hit ratios for the moving paths by all mobile users under schemes DF, SD-local, and SD-global can be obtained by using the function of the local access hit ratio described in Section 2. Note that the local hit ratios of mobile users using SD-local and SD-global are higher than that using DF. Also note that the infrequently moving users (such as U_2, U_3, and U_4) will have better local access hits when using SD-local than using SD-global. On the other hand, a frequently moving user like U_1 performs better under SD-global than under SD-local. These agree with our intuition in that SD-local deals with user occurrence counts and SD-global considers mainly movement occurrence counts, resulting in the situation that SD-local will favor infrequently moving users and SD-global is good for frequently moving users. In fact, these observations will be validated by our experimental studies in Section 4.

TABLE 4
The Scenarios under Different Shared Data Allocation Algorithms

User ID with an example moving path	Replicated Server under DF	Local hit ratio of DF
U_1 with CBAE	AFKP	$\frac{1}{4}f$
U_2 with BCGF		$\frac{1}{4}f$
U_3 with BCDH		0
U_4 with CGH		0

(a)

User ID with an example moving path	Replicated Server under SD-local	Local hit ratio of SD-local
U_1 with CBAE	BCGK	$\frac{2}{4}f$
U_2 with BCGF		$\frac{3}{4}f$
U_3 with BCDH		$\frac{2}{3}f$
U_4 with CGH		$\frac{2}{3}f$

(b)

User ID with an example moving path	Replicated Server under SD-global	Local hit ratio of SD-global
U_1 with CBAE	ABCE	f
U_2 with BCGF		$\frac{2}{4}f$
U_3 with BCDH		$\frac{2}{3}f$
U_4 with CGH		$\frac{1}{3}f$

(c)

(a) The scenario under DF, (b) the scenario under SD-local, and (c) the scenario under SD-global.

3.2 Analysis of SD-Local and SD-Global

In this section, we qualitatively analyze these shared data allocation algorithms derived above. Explicitly, we first derive the properties of SD-local and SD-global in Section 3.2.1. Then, we investigate the impact of the closeness measure on the quality of solutions obtained by SD-local and SD-global in Section 3.2.2.

3.2.1 Properties of SD-Local and SD-Global

Note that the objective function of SD-global can be further decomposed into the following equation:

$$\sum_{i=1}^{N} L(U_i) * n_i = f * \left(\sum_{n_i > P_T} \frac{|R \cap FS_i|}{|FS_i|} * n_i + \sum_{n_j < P_T} \frac{|R \cap FS_j|}{|FS_j|} * n_j \right).$$

In order to maximize the objective function of SD-global, SD-global includes those L_2 with a larger value of movement occurrence count to increase the value of $\sum_{n_i > P_T} \frac{|R \cap FS_i|}{|FS_i|} * n_i$. As a result, under SD-global, the local access hit ratio of frequently moving users is larger than that of infrequently moving users. According to the objective functions of local optimization and global optimization, we have the following property.

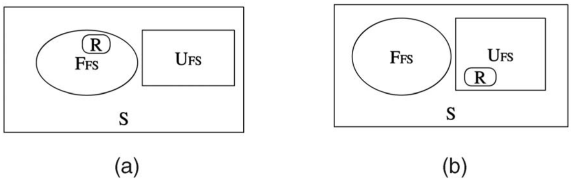

Fig. 5. Venn diagram of F_{FS} and U_{FS} when $C(F_{FS}, U_{FS}) = 0$.

Property 1. *SD-global is good for frequently moving users and SD-local favors infrequently moving users.*

The frequent sets of mobile users may overlap closely due to the moving behaviors of mobile users. In order to model the centrality for the moving behaviors of mobile users, we have the following definition.

Definition 3. *The clustering factor is defined as* $c_f = \frac{|S| - |\bigcup_{\forall i=1,\, 1 \le i \le N} FS_i|}{|S|}$*, where N is the total number of mobile users and* |S| *is the total number of servers.*

The larger value of c_f, the more centralized of moving behaviors for mobile users. With the larger value of c_f, the frequent set of mobile users overlap closely, which in turn causes the value of $\sum_{i=1}^{N} \frac{|R \cap FS_i|}{|FS_i|}$ to be increased. Also, note that a larger value of $\sum_{i=1}^{N} \frac{|R \cap FS_i|}{|FS_i|}$ is able to improve the objective functions of SD-local and SD-global. Thus, we have the following property.

Property 2. *The larger the value of* c_f*, the higher of local hit ratios of SD-local and SD-global will be.*

Note that these two properties of SD-local and SD-global will be verified by our experiments in Section 4.

3.2.2 The Impact of the Closeness Measure

With the formulation of the closeness measure described in Section 2, we have following three cases, i.e., $C(F_{FS}, U_{FS}) = 0$, $0 < C(F_{FS}, U_{FS}) < 1$, and $C(F_{FS}, U_{FS}) = 1$, to consider for analyzing the performance of SD-local and SD-global qualitatively.

Case 1: When $\mathbf{C(F_{FS}, U_{FS}) = 0}$. Fig. 5 shows the two possible scenarios under the condition $C(F_{FS}, U_{FS}) = 0$. In accordance with Property 1, SD-global is good for frequently moving users and SD-local favors infrequently moving users. Thus, in Fig. 5a, the set of R determined by SD-global is within F_{FS}, which is able to significantly increase the local access hit ratios of frequently moving users. In contrast, Fig. 5b shows that the set of R determined by SD-local is in U_{FS}, which could improve the local hit ratios of infrequently moving users.

Case 2: When $\mathbf{0 < C(F_{FS}, U_{FS}) < 1}$. Fig. 6 shows the scenarios under the condition $0 < C(F_{FS}, U_{FS}) < 1$. From Fig. 6a, since the whole R is within the set of F_{FS}, the R determined by SD-global can only improve the local hit ratios of frequently moving users. On the other hand, in Fig. 6b, SD-local is able to increase the local hit ratios of infrequently moving users due to that the whole R is enclosed in the set of U_{FS}. It can be seen by Figs. 6c and 6d that all mobile users can benefit by R since R is partially within the intersection set of F_{FS} and U_{FS}. Clearly, the specific location of R determines the degree of improvement obtainable. Explicitly, the improvement of frequently moving users in Fig. 6c is larger than that of infrequently moving users. In contrast, infrequently moving users can obtain more benefit than frequently moving users under the configuration shown in Fig. 6d. As shown in Fig. 6e, when the whole R falls in the intersection set of F_{FS} and U_{FS}, the local hit ratios of all mobile users will improve.

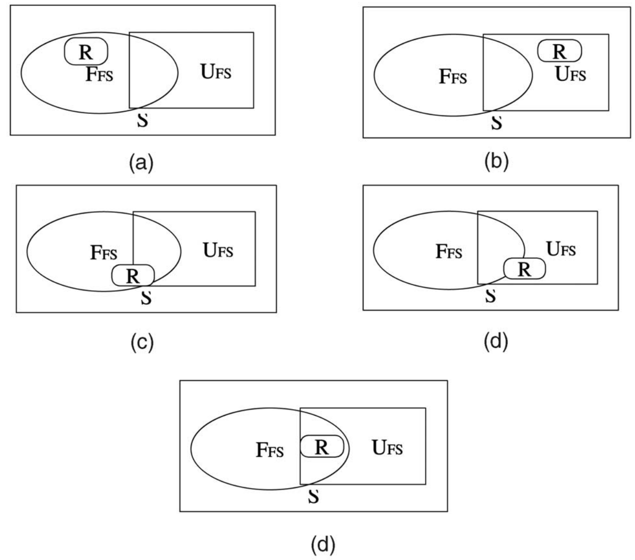

Fig. 6. Venn diagram of F_{FS} and U_{FS} when $0 < C(F_{FS}, U_{FS}) < 1$.

Case 3: When $\mathbf{C(F_{FS}, U_{FS}) = 1}$. Fig. 7 shows the scenario where $\mathrm{C}(F_{FS}, U_{FS}) = 1$, meaning that the set of F_{FS} is exactly the same as the set of U_{FS}. The set of R determined by SD-local and SD-global is within the set of F_{FS} and that of U_{FS} and is beneficial for both the frequently moving and those infrequently moving users. As a consequence, the difference of SD-local and SD-global under such condition, i.e., $C(F_{FS}, U_{FS}) = 1$, is expected to be small.

In all, the closeness measure, $C(F_{FS}, U_{FS})$, has significant influences on the solution quality obtainable of SD-local and SD-global. This will again be validated by our experiments in Section 4.3.1.

4 Performance Study

The effectiveness of using moving patterns for data allocation is evaluated empirically in this section. The simulation model is described Section 4.1. In Section 4.2, we examine the impact of employing user moving patterns for shared data allocation. Performance of SD-local and SD-global is comparatively analyzed in Section 4.3.

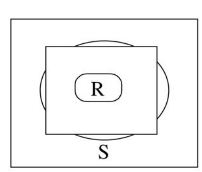

Fig. 7. Venn digram of F_{FS} and U_{FS} when $C(F_{FS}, U_{FS}) = 1$.

TABLE 5
The Parameters Used in the Simulation

Notation	Definition
N	Total number of mobile users
$C(F_{FS}, U_{FS})$	The closeness measure of F_{FS} and U_{FS}
S	Total number of servers in a mobile computing system
$\|R\|$	Number of replicated servers
SITEOP	Average number of operations performed in a server
L_{ave}	Average length of a moving path
P_{back}	Backward probability for user movement
c_f	Clustering factor

4.1 Simulation Model

Table 5 summarizes the definitions of primary simulation parameters. The number of mobile users in a mobile computing system is denoted by N and the number of servers in a mobile computing system is S. The closeness measure of F_{FS} and U_{FS} is expressed by $C(F_{FS}, U_{FS})$. A moving path is a sequence of servers accessed by a mobile user and the length of each moving path is modeled as a uniform distribution between $L_{ave} - 2$ and $L_{ave} + 2$. The same as in [13], [25], the moving behavior of mobile users is based on a roundtrip model, where the starting position of a moving path for a mobile user can be either VLR or HLR. The starting position of a moving path is randomly selected between 1 and $S * (1 - c_f)$ in each run of simulation, where c_f is the clustering factor. As pointed out before, the larger value of c_f, the more centralized of moving behaviors for mobile users. The number of operations submitted by a mobile user to its nearby server is modeled by a uniform distribution between SITEOP-20 and SITEOP+20, where SITEOP is assumed to be 50 in our experiments. After the server has completed these operations, the mobile user moves to one of its neighboring servers depending on a probabilistic model. Explicitly, the probability that a mobile user moves to the server where this user came from is modeled by P_{back} and the probability that the mobile user moves to the other servers is determined by $(1 - P_{back})/(n - 1)$ where n is the number of possible servers this mobile user can move to. The average local hit ratio for a user (henceforth referred to as local hit ratio), i.e.,

$$\frac{\sum_{i=1}^{N} L(U_i)}{N},$$

means the percentage that among all data accesses, data can be obtained from local servers of mobile users. For comparison purposes, scheme DF, which allocates data in a fix pattern, is implemented to randomly generate the set of replicated servers.

4.2 The Impact of Employing User Moving Patterns for Shared Data Allocation

To show the advantage of utilizing user moving patterns for shared data allocation, we set the value of $C(F_{FS}, U_{FS})$ to 0.2, the value of c_f to 0.5, the value of L_{ave} to 6, and the value of $|R|$ to 4. The local hit ratios of DF, SD-local and SD-global with the number of mobile users varied is shown in Fig. 8. As can be seen from Fig. 8, SD-local and SD-global significantly outperform DF, showing the advantage of utilizing moving patterns for shared data allocation. Fig. 9 shows the local hit ratios of DF, SD-local and SD-global with the number of servers varied. Note that since the number of replicated servers is set to 4 while the number of servers increases, the local hit ratios of DF, SD-local and SD-global tend to decrease. Due to the optimization criterion of SD-local described in Section 3.1, the local hit ratio of SD-local is larger than that of SD-global.

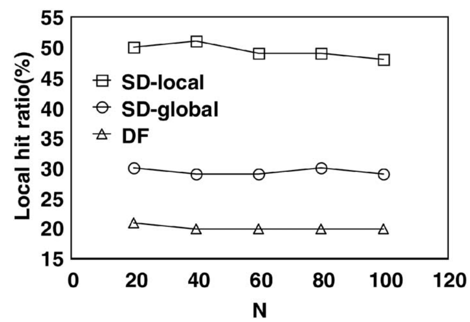

Fig. 8. The local hit ratios of DF, SD-local, and SD-global with the number of mobile users N varied.

It is seen from Fig. 9 that SD-local and SD-global can not only increase the local hit ratios of mobile users, but also is of practical use in large scale mobile computing systems.

4.3 Comparative Analysis of SD-Local and SD-Global

Performance of SD-local and SD-global is comparatively analyzed in this section. First, we examine the impact of varying the closeness measure. Then, the experiments of clustering factor c_f are conducted.

4.3.1 Experiments of Closeness Measure

We now examine the impact of varying the value of $C(F_{FS}, U_{FS})$ to the performance of SD-local and SD-global. Without loss of generality, we set the value of L_{ave} to 6, the value of S to 20, the value of N to 20, the value of $|R|$ to 4, and the value of c_f to 0.5. The performances of SD-local and SD-global with the value of $C(F_{FS}, U_{FS})$ varied are shown in Figs. 10 and 11.

Fig. 10 shows the local hit ratios of SD-local and SD-global with the value of $C(F_{FS}, U_{FS})$ varied, and Fig. 11 shows the total hit counts of SD-local and SD-global with the value of

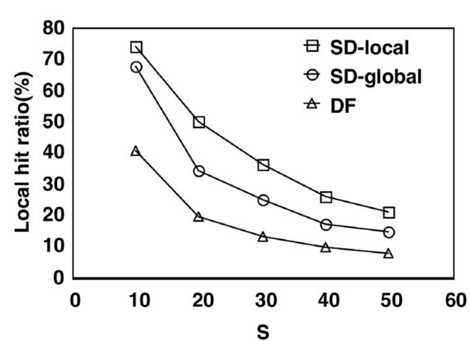

Fig. 9. The local hit ratios of DF, SD-local, and SD-global with the number of servers S varied.

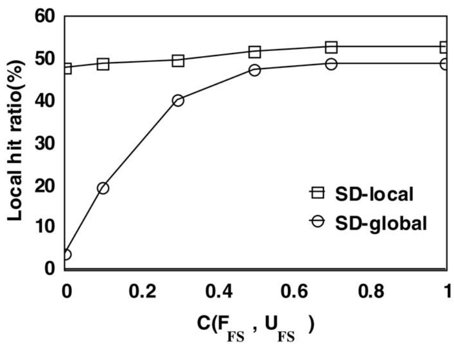

Fig. 10. The local hit ratios of SD-local and SD-global with $\mathrm{C}(F_{FS}, \mathrm{U}_{FS})$ varied.

$\mathrm{C}(F_{FS}, U_{FS})$ varied. As shown in Fig. 10, the local hit ratio of SD-local is larger than that of SD-global, showing the results from having different replicated servers employed by SD-local and SD-global. It is worth mentioning that although the local hit ratio of SD-local is larger than that of SD-global, the total hit count of SD-global is larger than that of SD-local in Fig. 11, showing the very difference in these two optimizations criteria described in Section 3.1.2. Note that SD-global achieves the global optimization in that the total hit count under SD-global is large than that of SD-local despite the average local hit ratios of mobile users under SD-local are larger than those under SD-global. Furthermore, it can be seen from Fig. 10 that as the value of $\mathrm{C}(F_{FS}, U_{FS})$ increases, the difference between SD-local and SD-global becomes smaller. This can be explained by the fact that more sites in R fall into the intersection set of F_{FS} and U_{FS} as $\mathrm{C}(F_{FS}, U_{FS})$ increases, which in turn improves the local hit ratios of both SD-local and SD-global. This agrees with our analysis in Section 3.2.

A complete spectrum for the local hit ratios of frequently moving users and infrequently moving users under algorithm SD-local is shown in Fig. 12 and that under algorithm SD-global is shown in Fig. 13. As can be seen in Fig. 12, the local hit ratio of infrequently moving users is larger than that of frequently moving users under algorithm SD-local. In contrast, Fig. 13 shows that the local hit ratio of frequently moving users is larger than that of infrequently moving users under SD-global. This also agrees with Property 1 mentioned before which states that SD-local will favor infrequently moving users and SD-global is good for frequently moving users. Clearly, the choice of SD-global and SD-local will be a design issue that is dependent on the system objective.

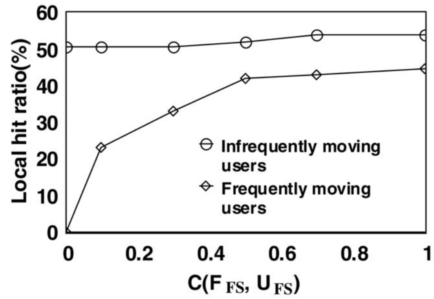

Fig. 12. The local hit ratios of frequently moving users and infrequently moving users under SD-local by varying $\mathrm{C}(\mathrm{F}_{FS}, \mathrm{U}_{FS})$.

4.3.2 Experiments of Clustering Factor

As described before, the centrality of frequent sets of mobile users affects the performance of SD-local and SD-global. To conduct the experiments to evaluate SD-local and SD-global with the value of c_f varied, we set the value of L_{ave} to 6, the value of $\mathrm{C}(F_{FS}, U_{FS})$ to 0.2, the value of S to 20, the value of $|\mathrm{R}|$ to 4 and the value of N to 20. The local hit ratios of SD-local and SD-global with various values of c_f are shown in Fig. 14, where it can be observed that a larger value of c_f will increase the value of $\sum_{i=1}^{N} \frac{|R \cap FS_i|}{|FS_i|}$, which in turn increases the objective functions of SD-local and SD-global. As a result, the larger the value of c_f, the higher of local hit ratios of SD-local and SD-global will be, agreeing with Property 2.

5 Conclusions

In this paper, we devised data allocation schemes that utilize the knowledge of user moving patterns for proper allocation of shared data in a mobile computing system. Specifically, by exploring the features of local optimization and global optimization, we derived the objective functions of local optimization and global optimization. With the objective functions, we devised algorithm SD-local and algorithm SD-global to achieve local optimization and global optimization, respectively. A measurement, called closeness measure which corresponds to the amount of

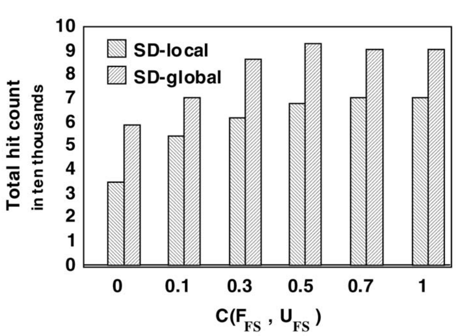

Fig. 11. The total hit counts of SD-local and SD-global with $\mathrm{C}(F_{FS}, \mathrm{U}_{FS})$ varied.

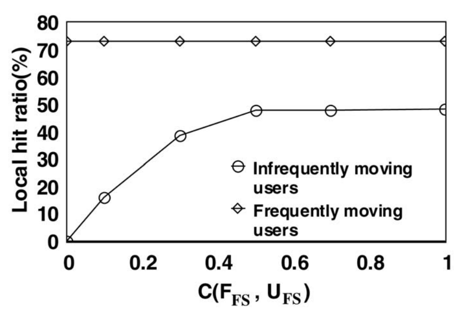

Fig. 13. The local hit ratios of frequently moving users and infrequently moving users under SD-global by varying $\mathrm{C}(\mathrm{F}_{FS}, \mathrm{U}_{FS})$.

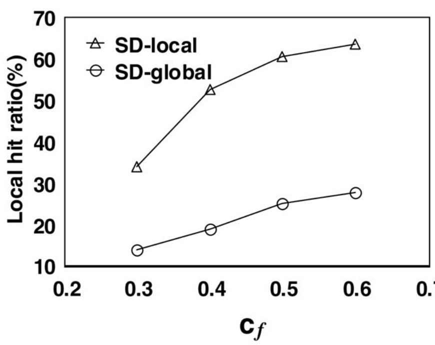

Fig. 14. The local hit ratios of SD-local and SD-global with the value of c_f varied.

the intersection between the set of frequently moving user patterns and that of infrequently moving user patterns, was derived to assess the quality of solutions resulted by SD-local and SD-global. Sensitivity analysis on various parameters was conducted and performance of those data allocation algorithms was comparatively analyzed. From the analysis of SD-local and SD-global, SD-local favors infrequently moving users and SD-global is good for frequently moving users. It was shown by our simulation results that the knowledge obtained from the user moving patterns is very important in devising effective shared data allocation algorithms which can lead to prominent performance improvement in a mobile computing system.

Appendix A

Mining for Moving Patterns in a Mobile Environment

User moving patterns refer to the areas in which users frequently travel. As mentioned before, similarly to the calling patterns of users, it is envisioned that most users tend to have their own user moving patterns since the behaviors of users are likely to be regular [5], [11], [25], [29]. In [4], [20], a solution procedure, which is composed of a sequence of algorithms, is developed to capture the user moving patterns from a set of log data in a mobile computing system.

In a mobile environment each mobile user is associated with a home location database which maintains an up-to-date location data for the mobile user. The location management procedure for a mobile computing system considered in this paper is similar to the one in IS-41/GSM [10], [17], which is a two level standard and uses a two-tier system of home location register (to be referred to as HLR) and visitor location register (to be referred to as VLR) databases. In order to capture user moving patterns, a moving pair, (old VLR, new VLR), is generated in a moving log for each registration procedure and then we can obtain a moving sequence $\{(O_1, N_1), (O_2, N_2), \dots (O_n, N_n)\}$ for each mobile user.

Once the movement log is generated, we shall convert the log data into multiple subsequences, each of which represents a *maximal moving sequence*. After maximal moving sequences are obtained, we then map the problem of finding frequent moving patterns into the one of finding frequent occurring consecutive subsequences among maximal moving sequences. A sequence of k movements is called a *large k-moving sequence* if there are a sufficient number of maximal moving sequences containing this k-moving sequence. Such a threshold number is called a support in this paper. Note that after large moving sequences are determined, moving patterns can then be obtained in a straightforward manner. A moving pattern is a large moving sequence that is not contained in any other moving patterns. For example, suppose that $\{\mathrm{AB}, \mathrm{BC}, \mathrm{AE}, \mathrm{CG}, \mathrm{GH}\}$ is the set of large 2-moving sequences and $\{\mathrm{ABC}, \mathrm{CGH}\}$ is the set of large 3-moving sequences. Then, the resulting user moving patterns are $\{\mathrm{AE}, \mathrm{ABC}, \mathrm{CGH}\}$. The overall procedure for mining moving patterns [20] is outlined below.

Procedure for mining of moving patterns

Step 1. **(Data collection phase)** Employing algorithm MM to determine maximal moving sequences from a set of log data and also the occurrence count of moving pairs.

Step 2. **(Mining phase)** Employing algorithm LM to determine large moving sequences for every w maximal moving sequences obtained in Step 1, where w is the retrospective factor which is an adjustable window size for the recent moving patterns to be considered.

Step 3. **(Pattern generation phase)** Determine user moving patterns from large moving sequences obtained in Step 2.

In order not to distract the readers from the main theme of this paper for shared data allocation, we do not include detailed mining algorithms to obtain user moving patterns. Interested readers are referred to [4], [20].

Acknowledgments

This work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE.

References

[1] D. Barbara, "Mobile Computing and Databases—A Survey," *IEEE Trans. Knowledge and Data Eng.*, vol. 11, no. 1, pp. 108-117, Jan./Feb. 1999.

[2] B. Bruegge and B. Bennington, "Applications of Mobile Computing and Communication," *IEEE Personal Comm.*, pp. 64-71, Feb. 1996.

[3] K. Buchanan, R. Fudge, D. McFarlane, T. Phillips, A.A. Sasaki, and H. Xia, "IMT-2000: Service Provider's Perspective," *IEEE Personal Comm.*, vol. 4, no. 4, pp. 8-13, Aug. 1997.

[4] M.-S. Chen, J.-S. Park, and P.S. Yu, "Efficient Data Mining for Path Traversal Patterns," *IEEE Trans. Knowledge and Data Eng.*, vol. 10, no. 2, pp. 209-221, Apr. 1998.

[5] G. Cho and L.F. Marshall, "An Efficient Location and Routing Scheme for Mobile Computing Environments," *IEEE J. Selected Areas in Comm.*, vol. 13, no. 5, pp. 868-879, June 1995.

[6] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithm.* MIT Press, 1990.

[7] E. Dahlman, B. Gudmundson, M. Nilsson, and A. Skold, "UMTS/IMT-2000 Based on Wideband CDMA," *IEEE Comm. Magazine*, vol. 36, no. 9, pp. 70-80, Sept. 1998.

[8] N. Davies, G.S. Blair, K. Cheverst, and A. Friday, "Supporting Collaborative Application in a Heterogeneous Mobile Environment," *Computer Comm.*, specical issue on mobile computing, 1996.

[9] M.H. Dunham, A. Helal, and S. Balakrishnan, "A Mobile Transaction Model That Captures Both the Data and Movement Behavior," *ACM J. Mobile Networks and Applications,* vol. 2, pp. 149-162, 1997.

[10] EIA/TIA, Cellular Radio Telecomm. Intersystem Operations, 1991.

[11] T. Imielinski and B.R. Badrinath, "Mobile Wireless Computing," *Comm. ACM,* vol. 37, no. 10, pp. 18-28, Oct. 1994.

[12] Intelligent Transportation Systems, http://www.artimis.org/, 2004.

[13] J. Jannink, D. Lam, N. Shivakumar, J. Widom, and D. Cox, "Efficient and Flexible Location Management Techniques for Wireless Communication Systems," *ACM J. Wireless Networks,* vol. 3, no. 5, pp. 361-374, 1997.

[14] J. Jing, A. Helal, and A. Elmagarmid, "Client-Server Computing in Mobile Environments," *ACM Computing Surveys,* vol. 31, no. 2, pp. 117-157, June 1999.

[15] D.N. Knisely, S. Kumar, S. Laha, and S. Nanda, "Evolution of Wireless Data Services: IS-95 to cdma2000," *IEEE Comm. Magazine,* pp. 140-149, Oct. 1998.

[16] N. Krishnakumar and R. Jain, "Escrow Techniques for Mobile Sales and Inventory Applications," *ACM J. Wireless Network,* vol. 3, no. 3, pp. 235-246, July 1997.

[17] Y.-B. Lin, "GSM Network Signaling," *ACM Mobile Computing and Comm.,* vol. 1, no. 2, pp. 11-16, 1997.

[18] L.B. Milstein, "Wideband Code Division Multiple Access," *IEEE J. Seclected Areas in Comm.,* vol. 18, no. 8, pp. 1344-1355, Aug. 2000.

[19] M. Nicola and M. Jarke, "Performance Modeling of Distributed and Replicated Databases," *IEEE Trans. Knowledge and Data Eng.,* vol. 12, no. 4, pp. 645-672, July 2000.

[20] W.-C. Peng and M.-S. Chen, "Mining User Moving Patterns for Personal Data Allocation in a Mobile Computing System," *Proc. 29th Int'l Conf. Parallel Processing (ICPP 2000),* Aug. 2000.

[21] E. Pitoura and G. Samaras, "Locating Objects in Mobile Computing," *IEEE Trans. Knowledge and Data Eng.,* vol. 13, no. 4, pp. 571-592, July/Aug. 2001.

[22] A. Samukic, "UMTS Universal Mobile Telecommunications System: Development of Standards for the Third Generation," *IEEE Trans. Vehicular Technology,* vol. 47, no. 4, pp. 1099-1104, Nov. 1998.

[23] M. Satyanarayanan, "Mobile Information Access," *IEEE Personal Comm.,* pp. 26-33, Feb. 1996.

[24] S. Shekhar, A. Fetterer, and D. Lui, "Genesis: An Approach to Data Disemination in Advanced Travel Information Systems," *IEEE Data Eng. Bull.,* vol. 19, no. 3, pp. 37-45, 1996.

[25] N. Shivakumar, J. Jannink, and J. Widom, "Per-User Profile Replication in Mobile Environments: Algorithms, Analysis and Simulation Results," *ACM J. Mobile Networks and Applications,* vol. 2, pp. 129-140, 1997.

[26] U. Varshney and R. Vetter, "Emerging Mobile and Wireless Networks," *Comm. ACM,* vol. 43, no. 6, pp. 73-81, June 2000.

[27] O. Wolfson, S. Jajodia, and Y. Huang, "An Adaptive Data Replication Algorithm," *ACM Trans. Database Systems,* vol. 22, no. 4, pp. 255-314, June 1997.

[28] A. Wolski, "Database Replication for the Mobile Era," *Tutorial of the Proc. 18th Int'l Conf. Data Eng.,* Feb. 2002.

[29] H.-K. Wu, M.-H. Jin, J.-T. Horng, and C.-Y. Ke, "Personal Paging Area Design Based on Mobile's Moving Behaviors," *Proc. IEEE Infocom 2001,* pp. 21-30, Apr. 2001.

Wen-Chih Peng received the BS and MS degrees from National Chiao Tung University, Taiwan, in 1995 and 1997, respectively, and the PhD degree in electrical engineering from the University of National Taiwan University, Taiwan, R.O.C in 2001. During his PhD program, he was mainly involved in the projects related to mobile computing, telelcommunication databases, and data mining. Since September 2003, he has been an assistant professor at National Chiao Tung University and his research interests include mobile computing, mobile data management, and data mining. He is a member of the Phi Tau Phi scholastic honor society.

Ming-Syan Chen received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, and the MS and PhD degrees in computer, information, and control engineering from The University of Michigan, Ann Arbor, in 1985 and 1988, respectively. Dr. Chen is currently a professor and the chairman of Graduate Institute of Communication Engineering, a professor in the Electrical Engineering Department, and also a professor in the Computer Science and Information Engineering Department, National Taiwan University, Taipei, Taiwan. He was a research staff member at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, from 1988 to 1996. His research interests include database systems, data mining, mobile computing systems, and multimedia networking, and he has published more than 180 papers in his research areas. In addition to serving as program committee members in many conferences. Dr. Chen served as an associate editor of *IEEE Transactions on Knowledge and Data Engineering* from 1997 to 2001, is currently on the editorial board of the *Very Large Data Base Journal*, *Knowledge and Information Systems Journal*, the *Journal of Information Science and Engineering*, and the *International Journal of Electrical Engineering*, and was a Distinguished Visitor of the IEEE Computer Society for Asia-Pacific from 1998 to 2000. He served as the program chair of PAKDD-02 (Pacific Area Knowledge Discovery and Data Mining), program vice-chair of IEEE International Conference on Distributed Computing Systems (ICDCS) 2005, International Conference on Parallel Processing (ICPP) 2003, program vice-chair of VLDB-2002 (Very Large Data Bases), and also general chair and program chairs of several other conferences. He was a keynote speaker on Web data mining at the International Computer Congress in Hong Kong, 1999, a tutorial speaker on Web data mining in DASFAA-1999, and on parallel databases in the 11th IEEE ICDE in 1995, and also a guest coeditor for *IEEE Transactions on Knowledge and Data Engineering* on a special issue for data mining in December 1996. He holds, or has applied for, 18 US patents and seven ROC patents in the areas of data mining, Web applications, interactive video playout, video server design, and concurrency and coherency control protocols. He is a recipient of the NSC (National Science Council) Distinguished Research Award and the K.-T. Li Research Penetration Award for his research work, and also the Outstanding Innovation Award from IBM Corporate for his contribution to a major database product. He also received numerous awards for his research, teaching, inventions, and patent applications. Dr. Chen is a fellow of the IEEE and the IEEE Computer Society, and a member of the ACM.