Allocation of Shared Data based on Mobile User Movement

Wen-Chih Peng and Ming-Syan Chen
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, ROC
E-mail:{mschen@cc.ee.ntu.edu.tw, wcpeng@arbor.ee.ntu.edu.tw}

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. By exploring the corresponding features, we devise algorithm SD-local and algorithm SD-global to achieve local optimization and global optimization, respectively. 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.

1 Introduction

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 [1][8]. For example, some emerging wireless corporate applications, such as sales force automation in which salesperson can instantly obtain the latest pricing and competitive information for their customers, access many data commonly used by many mobile users. These commonly used data are examples of shared data (referring to those data to be accessed by a group of users) considered in this paper.

For cost-performance reasons, a mobile computing system is usually of a distributed server architecture [1][3][5], 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 [5]. Data objects are assumed to be stored at servers to facilitate coherency control and also for memory saving at mobile units [7][9]. 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 [4][7][9]. 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.

It is noted that various data allocation schemes have been extensively studied in the literature [7][9]. 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 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 Figure 1. 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 Figure 1, the solid line

Proceedings of the Third International Conference on Mobile Data Management (MDM'02)
0-7695-1500-2/02 $17.00 © 2002 IEEE

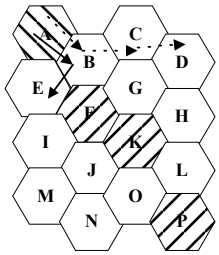

Figure 1. An example scenario of shared data allocation problem.

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 [7][10]. 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 corresponding features, 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. 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. 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.

This paper is organized as follows. Problem formulation is 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

In this paper, we devise data allocation algorithms that can utilize user moving patterns for proper allocation of shared data. 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:

Problem of shared data allocation based on user moving patterns: Given the number of mobile users with their moving patterns, 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. Table 1 shows the description of symbols used in modelling the problem. Figure 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 propositional to $\frac{|R \cap FS_i|}{|FS_i|}$, which is formulated as follows:

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

Consider the mobile user U_1 in Table 2, where the network topology is shown in Figure 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

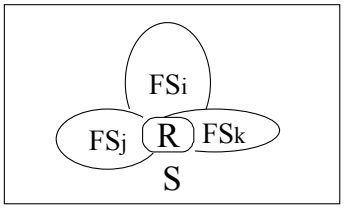

Figure 2. The problem of shared data allocation where the number of mobile users is 3.

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 relplcated 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

Table 1. Description of symbols.

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.

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:

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

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

Table 2. An example profile for illustrating shared data allocation schemes.

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 =1500 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

First, we describe in Section 3.1 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) in Section 3.2.

3.1 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 [9], 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.

3.2 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,

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

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:

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

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. Recall that moving patterns of mobile users may contain different large k-moving sequences 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 [6]. 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'_ks into L'_2s and the allocation of shared data will be made in accordance with the occurrences of these L'_2s. 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.

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

Table 3. An example profile for the counting in SD-local and SD-global.

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

1. Determine, from all user moving patterns, user occurrence counts of all frequent L'_2s
2. **Repeat** Until |V| $\leq$ 0; /* V is the number of replicated servers yet to determine*/
3. **begin**
4. Include those L'_2s that have maximal user occurrence count from all L'_2s into the set c-max.
   Also, c denotes an L_2 pair in c-max.
5. **if** |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. |V|=|V|-2;
10. **end**
11. **else if** ($\exists$ c $\in$ c-max and R$\cap$c $\neq$ 0)
12. **begin**
13. In c-max, choose an L_2 pair that has an intersection with pairs in R;
14. |V|=|V|-1;
15. **end**
16. **else** /* In c-max, there is no L_2 pair that has an intersection with pairs in R */

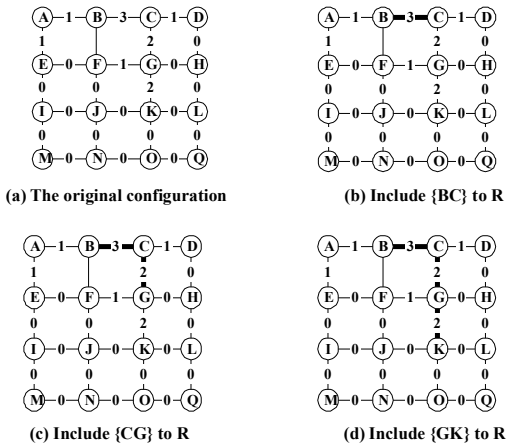

Figure 3. An execution scenario of algorithm SD-local.

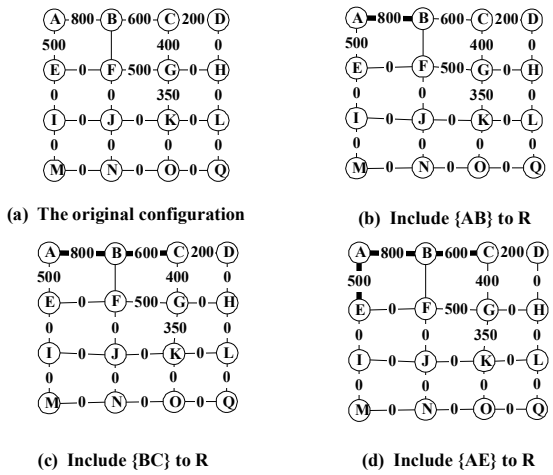

Figure 4. The execution scenario of algorithm SD-global.

17. **begin**
18. Choose an L_2 pair from c-max;
19. |V|=|V|-2;
20. **end**
21. R=R∪c;
22. **end**

end

Consider the execution of SD-local as an example shown in Figure 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 Figure 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., {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 Figure 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 {CG} is hence selected. This step is similar to Prim's algorithm for finding minimal-cost-spanning-tree (MCST) [2]. 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 {CG}, R becomes {BCG} and Figure 3b in turn leads to Figure 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 Figure 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 {BC} in moving paths of mobile user U_i. The movement occurrence count of {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 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, Figure 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.

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

Table 4a. The scenario under DF.

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		$\frac{2}{3}f$

Table 4b. The scenario under SD-local.

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		$\frac{1}{3}f$

Table 4c. The scenario under SD-global.

Table 4. The scenarios under different shared data allocation algorithms.

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 [6], *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.

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

Table 5. The parameters used in the simulation.

4 Performance Study

The effectiveness of using moving patterns for data allocation is evaluated empirically in this section. The simulation model is described in 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.

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 $L_{ave} + 2$. Same as in [4][7], 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 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.

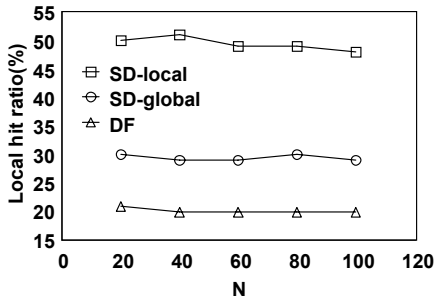

Figure 5. The local hit ratios of DF, SD-local and SD-global with the number of mobile users N varied.

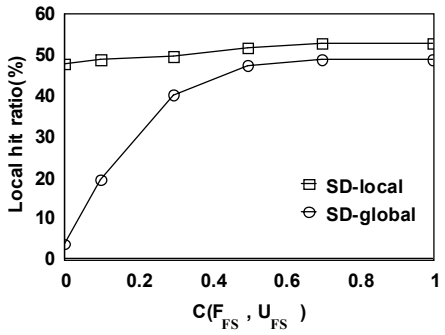

Figure 6. The local hit ratios of SD-local and SD-global with C(F_{FS}, U_{FS}) varied.

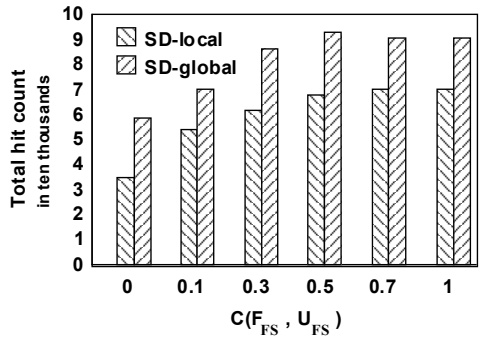

Figure 7. The total hit counts of SD-local and SD-global with C(F_{FS}, U_{FS}) varied.

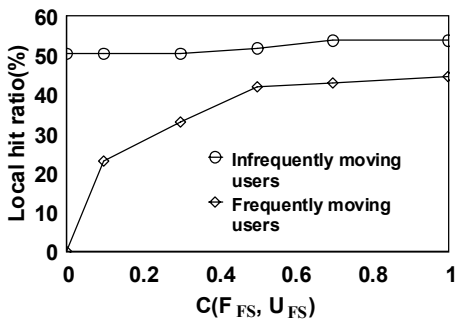

Figure 8. The local hit ratios of frequently moving users and infrequently moving users under SD-local by varying C($F_{FS,}$ U_{FS}).

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 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 Figure 5.

4.3 Performance of SD-local and SD-global

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 and the value of |R| to 4. The performances of SD-local and SD-global with the value of C(F_{FS}, U_{FS}) varied are shown in Figure 6 and in Figure 7.

Figure 6 shows the local hit ratios of SD-local and SD-global with the value of C(F_{FS}, U_{FS}) varied, and Figure 7 shows the total hit counts of SD-local and SD-global with the value of C(F_{FS}, U_{FS}) varied. As shown in Figure 6, 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 Figure 7, 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 Figure 6 that as the value of 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 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.

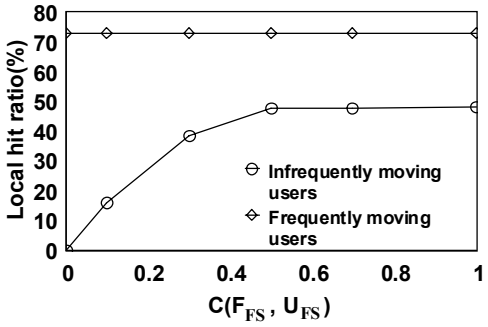

Figure 9. The local hit ratios of frequently moving users and infrequently moving users under SD-global by varying C(F_{FS}, U_{FS}).

A complete spectrum for the local hit ratios of frequently moving users and infrequently moving users under algorithm SD-local is shown in Figure 8 and that under algorithm SD-global is shown in Figure 9. As can be seen in Figure 8, the local hit ratio of infrequently moving users is larger than that of frequently moving users under algorithm SD-local. In contrast, Figure 9 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.

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 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.

ACKNOWLEDGEMENT

The authors are supported in part by the Ministry of Education Project No. 89-E-FA06-2-4-7 and the National Science Council, Project No. NSC 90-2213-E-002-086, Taiwan, Republic of China.

References

[1] D. Barbara. Mobile Computing and Databases–A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):108–117, January/February 1999.

[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithm. *MIT press*, 1989.

[3] M. H. Dunham, A. Helal, and S. Balakrishnan. A Mobile Transaction Model That Captures Both the Data and Movement Behavior. *ACM Journal of Mobile Networks and Applications*, 2:149–162, 1997.

[4] J. Jannink, D. Lam, N. Shivakumar, J. Widom, and D. Cox. Efficient amd Flexible Location Management Techniques for Wireless Communication Systems. *ACM Journal of Wireless Networks*, 3(5):361–374, 1997.

[5] N. Krishnakumar and R. Jain. Escrow Techniques for Mobile Sales and Inventory Applications. *ACM Journal of Wireless Network*, 3(3):235–246, July 1997.

[6] W.-C. Peng and M.-S. Chen. Mining User Moving Patterns for Personal Data Allocation in a Mobile Computing System. In *Proc. of the 29th International Conference on Parallel Processing (ICPP-2000)*, August 2000.

[7] N. Shivakumar, J. Jannink, and J. Widom. Per-User Profile Replication in Mobile Environments: Algorithms, Analysis and Simulation Results. *ACM Journal of Mobile Networks and Applications*, (2):129–140, 1997.

[8] K.-L. Tan and B. C. Ooi. Data Dissemination in Wireless Computing Environments. *Kluwer Academic Publishers*, 2000.

[9] O. Wolfson, S. Jajodia, and Y. Huang. An Adaptive Data Replication Algorithm. *ACM Transactions on Database Systems*, 22(4):255–314, June 1997.

[10] H.-K. Wu, M.-H. Jin, J.-T. Horng, and C.-Y. Ke. Personal Paging Area Design Based on Mobile's Moving Behaviors. In *Proceeding of IEEE Infocom 2001*, pages 21–30, April 2001.