Design and Performance Studies of an Adaptive Cache Retrieval Scheme in a Mobile Computing Environment

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

Abstract—In a mobile computing system, as users move to a new service area, the new server is usually considered to take over the execution of running programs for mobile users from the previous server so as to reduce the communication overhead of a mobile system. This procedure is referred to as service handoff. Note that when service handoff occurs, the new server will lose its advantage for cache access. To remedy this, we explore in this paper several cache retrieval schemes to improve the efficiency of cache retrieval. In particular, we analyze the impact of using a coordinator buffer to improve the overall performance of cache retrieval. Moreover, in light of the properties of transactions (i.e., temporal locality of data access among transactions), we devise a Dynamic and Adaptive cache Retrieval scheme (DAR) that can utilize proper cache methods according to some specific criteria to deal with the service handoff situation in a mobile computing environment. Performance of these cache retrieval schemes is analyzed and a system simulator is developed to validate our results. We devise a systematic procedure for determining the optimal operating points of DAR. Our experimental results show that by adaptively adopting the advantages of different cache retrieval methods, DAR significantly outperforms other schemes and is particularly effective for a mobile computing environment.

Index Terms—Mobile computing, service handoff, mobile database, cache retrieval scheme, temporal locality.

1 Introduction

In a mobile computing environment, a mobile user with a power-limited palm computer (or a mobile computer) can access various information via wireless communication [17], [23], [29], [30]. With a variety of data provided on the Internet, Internet access via wireless communication has become increasingly popular in recent years. For example, Advanced Traffic Information Systems (ATIS) [2], [13], [24] are systems developed to provide useful information for traveling. With portable computers, drivers are able to obtain updated information such as traffic reports, street maps, and entertaining information, to name a few. Traffic reports includes a variety of traffic data services such as the traffic routes with the most up-to-date traffic status, the shortest distance route, live traffic video, and location-dependent data. Though the prevalent service of mobile computing is voice service, it is believed that with the rapid advance in mobile streaming technology and content delivery networks for mobile computing, mobile streaming services are becoming widely available [8], [12], [20], [27]. Example system prototypes for mobile streaming services can be found in [27], [34].

It is noted that the computing capability and the storage size of palm computers are far less than those of desktop computers or servers. Thus, the mobile computing scenario is that mobile users submit their requests to the servers for processing. For a mobile application providing traffic routes, the traffic data stored at the central databases are continuously updated by the sensors and then retrieved by the mobile users. Such a scenario will unavoidably degrade the system performance and the quality of services. Hence, the mobile computing system for such streaming data services usually adopts a distributed server architecture [15], [17], [34], in which a service area refers to the converge area where the server can provide service to mobile users. In general, mobile users tend to submit transactions nearby so as to reduce the communication overhead incurred and also the response time of applications. The queries submitted by the mobile users are likely to incur long execution time due to the limitation of wireless bandwidth and the streaming nature of the data. As such, the mobile users may hop several service areas during the query execution time [7]. When mobile users enter a new service area, the new server is usually expected to take over from the previous server and continue the running applications seamlessly [15], [17]. This procedure is referred to as service handoff. Service handoff is important in that it reduces the communication cost between servers and mobile users. The response time to the mobile users can thus be minimized. In addition, service handoff can help balance workload of servers as well as increase the fault tolerance of a mobile computing system [15], [34].

Fig. 1 shows the computing scenario where there are n mobile users and data objects are assumed to be stored at the central databases to facilitate coherency control and also for storage saving at servers (i.e., in the coordinator databases in Fig. 1). The coordinator buffer is employed to share and cache those data frequently requested by all mobile users, thus reducing the number of disk access in the central databases. Specifically, in the end of the transaction

- *W.-C. Peng is with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, ROC. E-mail: wcpeng@csie.nctu.edu.tw.*
- *M.-S. Chen is with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, ROC. E-mail: mschen@cc.ee.ntu.edu.tw.*

Manuscript received 20 May 2002; revised 26 Mar. 2003; accepted 24 Sept. 2003; published online 1 Dec. 2004.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number 12-052002.

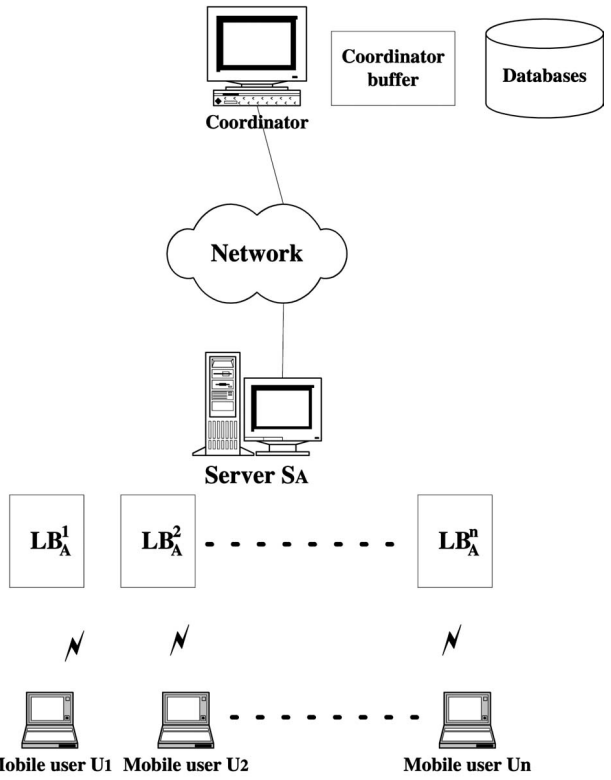

Fig. 1. The mobile computing environment considered in this paper.

execution, the transaction will do the coordinator buffer write which stores the updated data and invalidates obsolete data in the coordinator buffer. It has been reported that a coordinator buffer is useful in improving the system performance and scalability [5]. Due to the asymmetric feature of computing capability between servers and mobile computers [2], the mobile user submits the transaction to the server nearby and that server creates the corresponding local buffer to prefetch and cache those data requested. Denote the local buffer allotted for mobile user U_i in the service area of server S_A as LB_A^i. It can be seen that, for each mobile user, the server creates a server process in which the data cached are those frequently used data for the individual mobile user. As pointed out in [16], [34], such a computing scenario is also beneficial for maintaining cache consistency of mobile computers and providing scalable mobile streaming services. Once server S_A performs the operations of transactions issued by mobile user U_i, the server first checks LB_A^i for the data required. If the server does not find the data in LB_A^i, which is called a cache miss, a data request to the central databases is issued by the server. Clearly, caching recently accessed data in the server can significantly improve the performance of a mobile computing system. However, when the service handoff occurs and a new server takes over the running applications, the cache of the new server does not contain any data entry that was accessed by the newly taken transactions. The new server thus loses its advantage for cache access after service handoff. This is the very problem we shall address in this paper. To remedy this, we explore the technique of data prefetching for cache retrieval in a mobile computing system. Explicitly, before the transaction starts, fetching the data required to the local buffer will be able to reduce the response time of later access. This technique is referred to as prefetching. Thus, in order to maintain the use of cache after service handoff, the new server could prefetch those cache entries that are to be accessed by mobile users.

Three caching schemes are explored in this study. The first scheme is to access cache data in the server (to be referred to as FLB, standing for "from local buffer"). As pointed out earlier, when the mobile user submits transactions to the server, the server shall maintain the corresponding local buffer for that mobile user.[1] Second, one may utilize a coordinator buffer to cache data for the mobile computing system (to be referred to as FCB, standing for "from the coordinator buffer"). As mentioned above, the mobile computing system is of a distributed server architecture, in which data sharing can be achieved by employing a coordinator buffer to keep data to be shared by all servers [5], [9]. The third scheme is to prefetch cache data from the prior server of running transactions (to be referred to as FPS, standing for "from the previous server cache"). This scheme is included for our evaluation so as to capture the very nature of mobile computing.

The problem we should study can be best understood by the illustrative scenario in Fig. 2. In the beginning, served by server S_A, mobile user U_i submits a transaction to server S_A for traffic routes with the most up-to-date traffic status. Note that a transaction is a logical unit of mobile data services. In our illustrative traffic example, those data in the transaction submitted by mobile user U_i are related to traffic status captured by sensors. As can be seen in Fig. 2, server S_A creates the local buffer (i.e., LB_A^i) from which those data used by mobile user U_i are fetched. Since the transaction submitted by mobile user U_i just starts, LB_A^i does not contain any data entry. In order to utilize the advantage of cache, server S_A may prefetch those cache entries of traffic routes that are frequently accessed by mobile users. According to the transaction properties that will be discussed in details later, server S_A will determine whether the data should be prefetched to LB_A^i before the transaction is executed. Suppose that mobile user U_i moves to a new service area which is covered by server S_B. The running applications of mobile user U_i are then transferred to server S_B for execution. At the same time, server S_B also creates the local buffer (i.e., LB_B^i) for mobile user U_i. Obviously, LB_B^i does not have any data entry that was accessed by mobile user U_i. In order to maintain the use of cache, server S_B may *either* use one of the two schemes: FPS (i.e., from LB_A^i in server S_A) and FCB (i.e., from the coordinator buffer) to prefetch traffic data to LB_B^i, *or* simply employ FLB (i.e., from the local buffer) to avoid the communication overhead incurred by prefetching. Clearly, the employment of proper cache schemes has a significant impact to the system performance and should be determined in light of the transaction properties and execution efficiency. The design and analysis of a dynamic and adaptive cache retrieval scheme (referred to as DAR) that can utilize proper cache to deal with the service handoff situation in a mobile computing environment is the objective of this paper. Notice that the system prototype

1. Note that there is no data prefetching for this caching scheme.

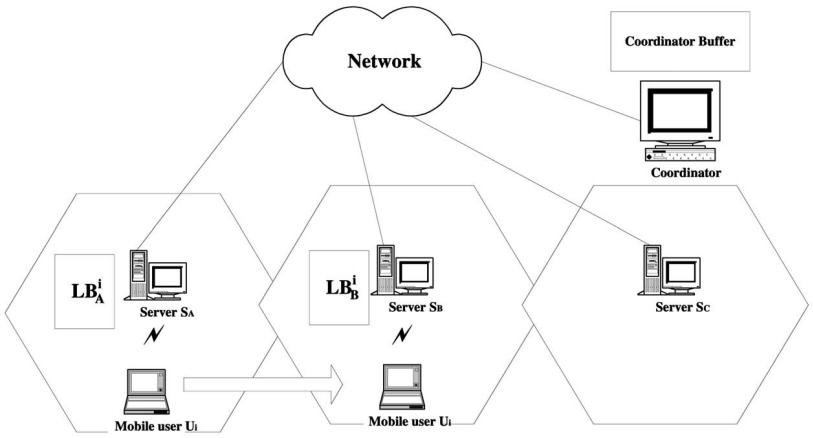

Fig. 2. A cache retrieval problem in a mobile computing system.

for mobile streaming services is being developed by NTT DoCoMo and HP, thereby justifying the importance of our study [27], [34].

The study on cache retrieval for a mobile computing system is different from that for a traditional database system not only in the related cost model but also due to the occurrence of service handoff. In this paper, we shall first evaluate the properties of some cache retrieval methods (i.e., FLB, FCB, and FPS), and then, in light of these properties, devise specific decision rules for DAR to employ proper caching methods to handle service handoff. A system simulator is built to validate our results. The effectiveness of the four caching retrieval schemes, i.e., FPS, FCB, FLB, and DAR is comparatively analyzed. It is found that temporal locality of transactions (which refers to the feature that consecutive transactions are likely to access same data [5], [9]) plays an important role for cache access in mobile computing. By taking into consideration the transaction properties and the costs of cache miss and cache replacement, DAR will select an appropriate method in each phase of transaction processing. Explicitly, let *intra-transaction page probability* mean the percentage of pages that demonstrates the intratransaction temporal locality and *intertransaction page probability* mean the percentage of pages that demonstrates the intertransaction temporal locality. The ratio of intratransaction page probability to intertransaction page probability, called the threshold ratio, is identified as a key parameter for the selection of appropriate methods in the execution of DAR. Sensitivity on the threshold ratio to the overall performance is analyzed. We devise a systematic procedure for determining the optimal operating points of DAR. DAR is adaptive in that for transactions with more intratransaction data, DAR has a higher cache hit ratio for intratransaction objects; whereas, for transactions with more intertransaction data, DAR has a higher cache hit ratio for intertransaction objects. This very advantage of DAR enables DAR to significantly outperform other schemes and is particularly effective for a mobile computing environment.

1.1 Related Work

Much research effort on caching has been elaborated upon caching at the proxy servers [1], [4], [10], [21], [25]. With the fast increase in mobile applications, cache management in a mobile computing system has been explored recently [2], [16], [18], [32], [33]. Our study in this paper distinguishes itself from others in that we focus on the cache retrieval to deal with service handoff in a mobile computing system.

The attention of those studies in [11], [14], [16], [31] was mainly paid to the cache invalidation scheme for the cache in a mobile unit. We mention in passing that Kahol et al. [16] and Pitoura and Bhargava [22] proposed some cache invalidation strategies and addressed the impact of disconnection time of clients to the overall performance. Wu et al. [31] proposed an energy-efficient cache invalidation scheme in which the cache invalidation will only take place if the cache data objects are frequently updated ones. Jing et al. [14] proposed an adaptive cache invalidation algorithm that uses adaptable mechanisms to adjust the size of the cache invalidation report so as to optimize the use of a limited amount of communication bandwidth.

Issues on cache granularity, coherence strategy, and replacement policy of mobile caching were investigated in [3], [6], [18], [28] via simulation models. Xu et al. [32] proposed an efficient gain-based cache replacement policy in which the influence of data size, data retrieval delay, access probability, and update frequency is considered together.

In addition, the use of a coordinator in the mobile computing system was pointed out in [17], [22] to coordinate the concurrency control scheme and to monitor the execution of transactions. However, their use of the coordinator is not for cache retrieval improvement.

To the best of our knowledge, prior work neither explicitly addressed the problem of cache retrieval to deal with service handoff for mobile computing nor considered the adaptive use of cache methods, let alone developing the corresponding decision rules and analyzing the impact of temporal locality of transactions to a mobile computing system. These features distinguish this paper from others.

This paper is organized as follows: Three basic cache retrieval methods (i.e., FLB, FCB, and FPS) are described in Section 2. The DAR scheme and the corresponding decision rules are developed in Section 3. A system simulator is

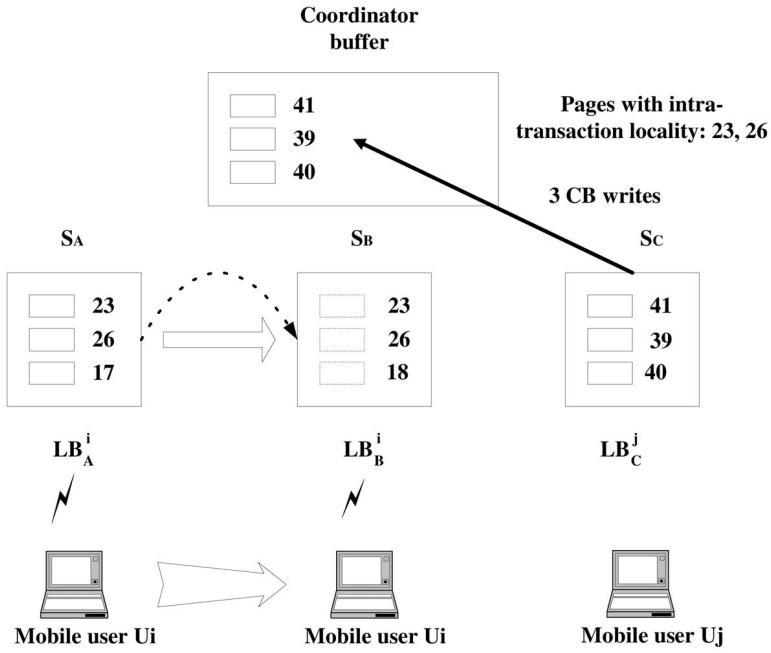

Fig. 3. Scenario of prefetching cache data from the previous server.

developed and various performance studies are conducted in Section 4. This paper concludes with Section 5.

2 Preliminaries

In a mobile computing system, mobile users submit transactions to the servers for execution, and the transactions request data pages from servers to process. In some applications, if data pages are referenced by transactions, these data pages have a tendency of being referenced again soon. This property is called temporal locality [5], [26]. For applications with temporal locality, the data pages can be further divided into two types of pages, namely, pages with intratransaction locality and pages with intertransaction locality [9]. Intratransaction locality refers to the feature that the same data pages are usually referenced within a transaction boundary, meaning that these pages present temporal locality within a transaction. In contrast, intertransaction locality refers to the feature that the same data pages are usually shared by some consecutive transactions. Consider the traffic report as an example. Transactions for querying up-to-date traffic routes might contain intertransaction pages (e.g., the traffic routes to different restaurants since queries for different restaurants are likely to be made consecutively) and intratransaction pages (e.g., the traffic route to the home address).

2.1 Description of Three Cache Retrieval Methods

We now describe three basic cache retrieval methods. As can be seen later, depending on the transaction properties, FLB, FPS, and FCB have their own advantages. Note that, after prefetching the cache data from the previous server or from the coordinator buffer, the server then accesses the cache data from the local buffer.

FLB (Caching from Local Buffer). As described before, for each mobile user, the server maintains the local buffer for each individual mobile user. In general, the server will create the local buffer and fetch data for each mobile user in the beginning of transaction execution. This is referred to as cache warm-up. Note that, when a mobile user enters the new service area, the new server will also create the local buffer for that mobile user so as to continue the running applications. At first, the local buffer created by the server does not contain any data entry until the transaction requests data from the central databases. Explicitly, no prefetching is performed before the transaction is executed. For the shortest distance route in the traffic report, those data of the shortest distance route do not present temporal locality since the query results mainly depend on the current location of mobile users. When the temporal locality is absent, FLB tends to perform better since FLB reduces the prefetch efforts. Without utilizing the technique of data prefetch, FLB is implemented and evaluated mainly for comparison purposes in Section 4.

FPS (Caching from Previous Server). For a transaction with higher intratransaction locality and lower intertransaction locality, obtaining the cache data from the previous server is effective for mobile computing. Fig. 3 illustrates such a scenario. For ease of description, a page with intratransaction (respectively, intertransaction) locality is referred to as an intratransaction (respectively, intertransaction) page. Consider the traffic report as an example. Mobile user U_i requires the traffic route to his/her home address. Those intratransaction pages, i.e., page 23 and page 26 in Fig. 3, have the information about the home address. Local buffer LB_A^i in server S_A contains the cache pages 23, 26, and 17 for the mobile user U_i, and the coordinator buffer contains the cache pages 41, 40, and 39 after the update[2] to the coordinator buffer by the mobile user U_j at server S_C. Suppose that, during the query execution time for traffic

2. Such an update is referred to as cross-invalidation in [5].

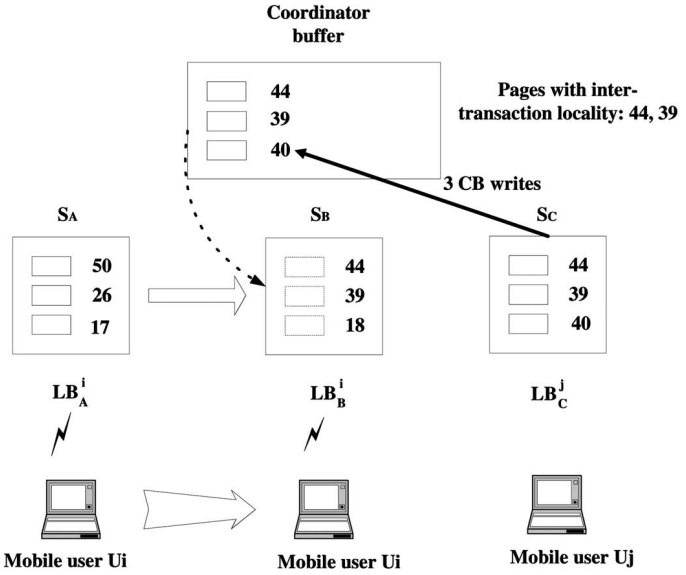

Fig. 4. Scenario of prefetching cache data from the corrdinator buffer.

routes, mobile user U_i moves into the service area of server S_B and is likely to request pages 23, 26, and 18. In our illustrative example, as service handoff occurs, prefetching the cache data from LB^i_A to LB^i_B will be more effective since LB^i_A contains more recently accessed intratransaction pages than the coordinator buffer. In general, FPS is favored for the case where transactions are of high intratransaction locality and low intertransaction locality.

FCB (Caching from Coordinator Buffer). If the transaction property is update-intensive and transactions possess higher intertransaction locality, obtaining cache data from the coordinator buffer will be cost-effective. Consider the traffic report for obtaining the available parking spaces in parking lots and each parking lot continuously updates the number of available parking spaces. Clearly, those data pages, i.e., pages 44 and 39 in Fig. 4, recording the number of available parking spaces are intertransaction pages. In Fig. 4, the coordinator buffer contains up-to-date pages 44, 39, and 40 after the update to the coordinator buffer by most recently committed transactions in server S_C. Assume that the mobile user U_i moves into the service area of server S_B and as usual, server S_B creates the local buffer LB^i_B not containing any data entry. Suppose that mobile user U_i is still interested in the number of available parking spaces after moving into the area of server S_B. As such, mobile user U_i is to access the data pages 44, 39, and 18 in LB^i_B. It can be seen in Fig. 4 that during the service handoff, prefetching cache data from the coordinator buffer to LB^i_B will have two cache hits, i.e., pages 44 and 39, and only incurs one cache miss, i.e., page 18, for later access of mobile user U_i. Clearly, FCB performs better than FPS in this case.

2.2 Three Phases of a Transaction

As pointed out earlier, DAR, the dynamic and adaptive cache retrieval scheme we shall devise in this paper will employ proper cache methods to deal with the service handoff situation, and a systematic procedure for determining the optimal operating points of DAR will be devised in Section 4. Fig. 5 shows the general cache retrieval scheme of DAR. In this general model, the transaction processing can be divided into three phases, namely, the initial phase, the execution phase and the termination phase.

During the initial phase, the transaction sets up the processing environment (explicitly, the transaction identification, local variables, and cache entry table are created). The cache entry table is created by the server, and two cache methods, FCB and FLB, are considered. Note that, since the transaction just started, FPS is not proper for this initial phase. FCB and FLB will be evaluated to decide which one to be used for the initial phase.

The second phase is the execution phase when the transaction is being processed by the server. If the server needs to do the service handoff when a mobile unit enters a new service area, the running transactions will migrate to a new server. The new server should then take over the running transactions seamlessly. As the new server sets up the running environment, cache data will be retrieved by the new server using three schemes: FLB, FCB, and FPS. We will evaluate these three schemes based on the corresponding transaction properties. The last phase of a transaction is the termination phase. In this termination phase, as the transaction execution finishes, the transaction will do the coordinator buffer write and activate the cache invalidation scheme to invalidate obsolete pages in the

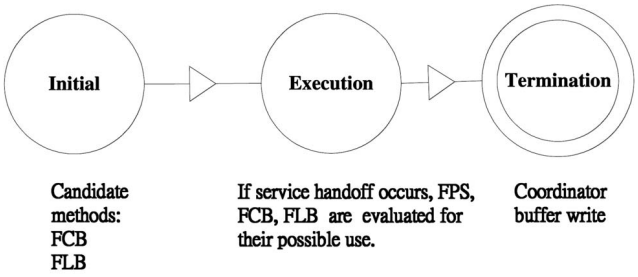

Fig. 5. A general cache retrieval scheme called Dynamic Adaptive Retrieval (DAR).

TABLE 1
Transaction Properties and Schemes Suggested in the Initial Phase

Temporal locality	Schemes suggested in the initial phase
+	FCB
–	FLB

Legends: +: The corresponding property is stronger in the transaction.
-: The corresponding property is weaker in the transaction.

TABLE 2
Transaction Properties and Schemes Suggested in the Execution Phase

Properties of transactions		Schemes suggested in the execution phase
Inter-transaction locality	Intra-transaction locality	
+	+	FPS/FCB
+	–	FCB
–	+	FPS
–	–	FLB

coordinator buffer. In essence, the dynamic and adaptive cache retrieval scheme (DAR) we devise will utilize the transaction properties and take the corresponding cost into consideration (i.e., cache miss and replacement) to evaluate effective cache retrieval methods in each transaction processing phase.

3 Dynamic and Adaptive Cache Retrieval Schemes

In this section, we shall first evaluate the performance of cache retrieval methods (i.e., FLB, FCB, and FPS), and then use the results obtained to devise DAR. Explicitly, cache retrieval methods for the initial phase are investigated in Section 3.1 and those for the execution phase are examined in Section 3.2. Decision rules for DAR are derived in Section 3.3.

3.1 Caching Schemes for the Initial Phase

Consider caching schemes for the initial phase. The transaction properties are taken into consideration to evaluate which scheme to use. Qualitatively speaking, when the transaction has temporal locality, FCB tends to perform better. Since with the temporal locality, the buffer in the coordinator maintains many intertransaction pages for different transactions and the server is thus likely to get pages from the coordinator buffer. On the other hand, when temporal locality is absent, FLB tends to perform better than FCB because that prefetching data from the coordinator buffer in FCB will incur more communication overhead. From the above reasoning, the schemes suggested for a transaction in the initial phase are shown in Table 1. Detailed criteria will be devised in Section 3.3.1. It is indicated by Table 1 that, when transactions have prominent temporal locality, FCB is favored. Otherwise, FLB is used. These properties will be validated by our experimental studies in Section 4.

3.2 Caching Schemes for the Execution Phase

We now consider caching schemes for a transaction in the execution phase. With a long transaction execution time, a transaction may migrate to a new server due to the movement of a mobile unit. It can be seen from the examples in Section 2 that temporal locality is a very important factor to be evaluated for determining which caching scheme to employ. Table 2 shows the schemes suggested for a transaction in the execution phase. It is indicated by Table 2 that, when intertransaction locality is prominent, FCB is the choice, and, when intratransaction locality is prominent, FPS should be used. On the other hand, similarly to that in the initial phase, if the transaction does not have the temporal locality, the FLB is used as a default scheme. The properties indicated in Table 2 will also be validated empirically in Section 4.

3.3 Deriving Decision Rules for DAR

By taking into consideration the transaction properties and the costs of cache miss and cache replacement, DAR will select an appropriate method in each phase of transaction processing. We shall conduct formula analysis and provide criteria for using DAR.

Intratransaction page probability, denoted by P_{intra}, represents the percentage of pages that demonstrates the intratransaction locality, whereas intertransaction page probability, denoted by P_{inter}, represents the percentage of pages that demonstrates the intertransaction locality. If a page exhibits both properties, we consider it as an intratransaction page. The sum of P_{intra} and P_{inter} is thus smaller than one. Recall that the intratransaction pages refer to those pages accessed by a single transaction, whereas the intertransaction pages are those accessed by different transactions. Similarly to the working-set model for calculating temporal locality [26], the values of P_{intra} and P_{inter} can be approximated from the statistics of prior data access. Assume that the number of cache entries for each mobile user is N. The threshold ratio of $\mathrm{P}_{intra}/\mathrm{P}_{inter}$, denoted by φ, is used for the selection of appropriate methods in the execution of DAR. The cache miss cost of each page and the probability for a page to possess temporal locality in the coordinator buffer are denoted by C_M and P_{CB}, respectively. Note that the value of P_{CB} is dependent upon several factors such as the characteristics of transactions, the coordinator buffer size, the replacement policy used in the coordinator buffer, and the frequency of coordinator buffer write incurred by transactions. Table 3 summarizes the descriptions of symbols used.

3.3.1 Decision Rule for the Initial Phase

The number of pages with temporal locality among data access of each mobile user can be expressed as $N * (P_{intra} + P_{inter})$. Then, we consider the cache miss cost for FCB and FLB. As pointed out earlier, in the beginning of transaction execution, the server will create the corresponding local buffer in which there are no data entry. Thus, the cost of cache miss for FLB is $C_M * N$. On the other hand, prefetching cache data from the coordinator buffer, FCB contains $N * (P_{intra} + P_{inter})$ cache data with temporal locality in which there are $N * (P_{intra} + P_{inter}) * P_{CB}$ pages appearing in the coordinator buffer. Note that the number of cache misses for the pages with temporal locality in FCB

TABLE 3
Description of Symbols

Description	Symbol
Number of cache entries in the server for each mobile user	N
Intra-transaction page probability	P_{intra}
Inter-transaction page probability	P_{inter}
Threshold ratio of $\frac{P_{intra}}{P_{inter}}$	φ
Cost of cache miss for each cache entry	C_M
Probability for a page to possess temporal locality in the coordinator buffer	P_{CB}

is $N * (P_{intra} + P_{inter}) * (1 - P_{CB})$ and the number of cache misses for the pages without temporal locality in FCB is $N * (1 - (P_{intra} + P_{inter}))$. As such, the cost of cache miss for FCB can be formulated as

$$N * (P_{intra} + P_{inter}) * (1 - P_{CB}) * 2 * C_M + N * (1 - (P_{intra} + P_{inter})) * 2 * C_M.$$

FCB is profitable in the initial phase if

$$N * (P_{intra} + P_{inter}) * (1 - P_{CB}) * 2 * C_M + N * (1 - (P_{intra} + P_{inter})) * 2 * C_M < C_M * N.$$

Formally, we have the following inequality to determine whether FCB or FLB should be used.

$$\begin{aligned} & N * (P_{intra} + P_{inter}) * (1 - P_{CB}) * 2 * C_M \\ & \quad + N * (1 - (P_{intra} + P_{inter})) * 2 * C_M < C_M * N \\ & => 2*((P_{intra}+P_{inter})*(1-P_{CB})+(1 - (P_{intra} + P_{inter})))<1 \\ & => \frac{0.5}{P_{CB}} < (P_{intra} + P_{inter}). \end{aligned}$$

Clearly, with a higher degree of temporal locality of transactions, the coordinator buffer will contain more frequently used pages with temporal locality since each transaction does the coordinator buffer write upon its completion. Using FCB to retrieve cache data will thus lead to a higher cache hit. However, if transactions do not have prominent temporal locality, using FLB to retrieve cache data is cost effective in that FCB will incur more communication overhead to achieve similar cache hit ratios as FLB. In brief, if $P_{intra} + P_{inter}$ is larger than $\frac{0.5}{P_{CB}}$, meaning that temporal locality is prominent, FCB is used. Otherwise, FLB is used. With several factors affecting the value of P_{CB}, the threshold value of $\frac{0.5}{P_{CB}}$ is determined empirically in Section 4. We have the decision rule for the initial phase as follows:

Decision Rule for the Initial Phase:
if ($P_{intra} + P_{inter} > \frac{0.5}{P_{CB}}$) **then**
    Using FCB.
else
    Using FLB.

3.3.2 Decision Rules for the Execution Phase

We now consider decision rules for the execution phase. Same as in the initial phase, if $P_{intra} + P_{inter}$ is less than $\frac{0.5}{P_{CB}}$, meaning that temporal locality is not prominent, FLB is used. However, if the transaction property has prominent temporal locality (i.e., $P_{intra} + P_{inter}$ is larger than $\frac{0.5}{P_{CB}}$), we shall select the schemes from FCB and FPS. In light of the temporal locality, we shall employ a control parameter, the temporal locality threshold φ, to decide which scheme to use. Explicitly, we use FPS if $P_{intra}/P_{inter} \geq \varphi$ (meaning that intratransaction locality is more prominent), and use FCB if $P_{intra}/P_{inter} < \varphi$ (meaning that intertransaction locality is more prominent). The value of φ will be determined empirically in Section 4 later. We have the decision rule for the execution phase as follows:

Decision Rule for the Execution Phase:
if ($P_{intra} + P_{inter} < \frac{0.5}{P_{CB}}$) **then**
    Using FLB.
else
    if $P_{intra}/P_{inter} \geq \varphi$ **then**
        Using FPS.
    else
        Using FCB.

Once the corresponding thresholds, i.e., $\frac{0.5}{P_{CB}}$ and φ, are determined for DAR, one can employ these decision rules derived in the initial phase and in the execution phase for the selection of proper cache methods.

4 Performance Study of Cache Retrieval Schemes

In Section 4.1, we describe our simulation model. The experimental results of the simulation are then discussed in Section 4.2. We analyze in Section 4.3 the impact of the coordinator buffer size to the value of φ, and describe a systematic procedure for determining the operating points of DAR. It will be seen that temporal locality has a significant impact to the cache retrieval methods and that DAR, due to its adaptability, significantly outperforms FLB, FPS, and FCB, rather than simply performing the same as the best of those three schemes.

4.1 Mobile System Simulation Model

In order to evaluate the performance of DAR, we develop a discrete event simulation model using SIMSCRIPT II.5. To simulate the information servers of the mobile information system, we use an 8x8 mesh topology network [19]. Each node in this 8x8 mesh topology represents one information server, and there are 64 information servers in this model. The arrival of each mobile user to each server is approximated by a Poisson process. The mobile user submits to the server several operations that are modeled as a uniform distribution between SITEOP-2 and SITEOP+2. After the server finishes these operations, the mobile user moves to one of the neighboring servers depending on a

TABLE 4
The Parameters Used in the Simulation

Notation	Definition	Value
Arrival	Num. of the customers arrive to each server	Exponential distri. with mean 10
SITEOP	Num. of operations performed in a server	Unifrom distri. with mean SITEOP
TXNSIZE	Num. of data objects needed in a transaction	20-30
DBSIZE	Num. of objects in the server	2600 objects
Server.Cache	Num. of cache entries for each mobile user	1% of DBSIZE
Coord.Cache	Num. of cache entry in the coordinator server	50%, 80%, 100% of DBSIZE
P_{intra}	Percentage of intra-transaction objects	Various values used
P_{inter}	Percentage of inter-transaction objects	Various values used
P_{pseudo}	Percentage of pseudo objects	Various values used

random routing function of the mesh network. The total number of data objects needed in a transaction is represented as TXNSIZE [9]. Table 4 summarizes the definitions and the values used for some primary simulation parameters. The size of the cache in each server is 1 percent of DBSIZE. Also, as in [3], [31], the size of the cache in the coordinator server is 50 percent of DBSIZE. Data objects in the cache are managed according to the LRU (Least Recently Used) replacement policy. To model the mobile streaming applications, the properties of data objects are mainly read-only in our simulation model.

It is assumed that there are K data objects in the database and there are three types of objects in the database, where various values of K were used in the experiments. The first type of objects is the intertransactions object (with a quantity of $P_{inter}{}^{*}K$ objects). The access to intertransaction objects is modeled by a uniform distribution with the range $[1, P_{inter}{}^{*}K]$. The second type is the intratransaction object (with a quantity of $P_{intra}{}^{*}K$ objects). In order to capture the nature of intratransaction locality, we divide the $P_{intra}{}^{*}K$ objects into $(P_{intra}{}^{*}K)/TXNSIZE$ groups. Each transaction will access intratransaction objects in its own group, and a normal distribution for page selection is employed to model the temporal locality of intratransaction. The third type is the pseudo object (with a quantity of $P_{pseudo}{}^{*}K$ objects) that are those data objects without temporal locality. Similarly, the access to pseudo object is modeled by a uniform distribution with the range $[1, P_{pseudo}{}^{*}K]$. As such, the total objects accessed by a transaction are composed of three parts, i.e., P_{intra} portion of intratransaction objects, P_{inter} portion of intertransaction objects, and $(1 - P_{intra} - P_{inter})$ portion of pseudo objects. In our experiments, a data page of 4K bytes is used as an object.

4.2 Experimental Result

The effectiveness of the four caching retrieval schemes, i.e., FPS, FCB, FLB, and DAR will be comparatively analyzed in this section. We first examine the impact of temporal locality to the cache hit ratios of FPS, FCB, and FLB in Section 4.2.1 and then evaluate the performance of DAR in Section 4.2.2.

4.2.1 The Impact of Temporal Locality

First, we evaluate the effect of varying the percentage of intratransaction pages (i.e., P_{intra}) accessed by transactions. Specifically, we fix the percentage of intertransaction (i.e., P_{inter}) to 10 percent, and examine the cache hit ratios when the value of P_{intra} increases. Experiments with different values of P_{inter} convey the same information and are thus omitted in this paper. Fig. 6 shows the resulting cache hit ratios of FPS, FCB, and FLB. It can be seen from Fig. 6 that FLB has the lowest cache hit ratio and the cache hit ratios of FPS and FCB increase prominently when P_{intra} increases. Moreover, the cache hit ratio of FPS is higher than that of FCB, due mainly to the fact that the cache replacement in the coordinator buffer is much more intensive. From these results, it is noted that FPS performs well for transactions with a large value of P_{intra}.

To provide more insights into this experiment, the distributions of intratransaction and intertransaction cache hit counts are shown in Fig. 7, from which it can be seen that in these three schemes the cache hit counts for intratransaction pages increase as P_{intra} increases (Recall that P_{inter} is fixed to 10 percent). This phenomenon is more

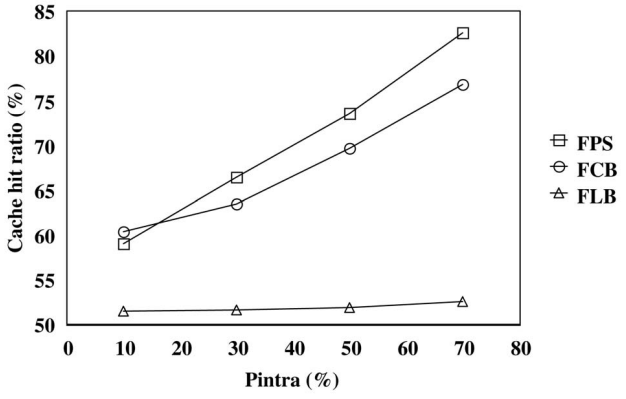

Fig. 6. The cache hit ratios of FPS, FCB and FLB by varying the value of P_{intra}.

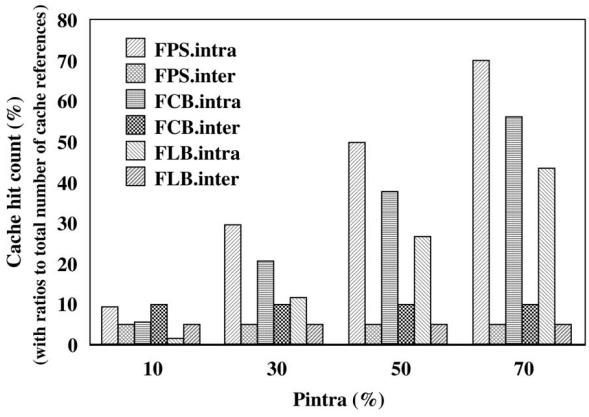

Fig. 7. The distributions of cache hit counts for intratransaction and intertransaction pages with P_{intra} varied.

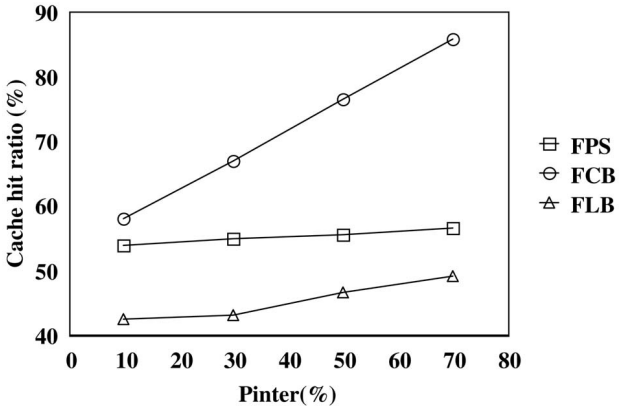

Fig. 8. The cache hit ratios of FPS, FCB, and FLB by varying the value of P_{inter}.

prominent for the case of FPS. In fact, FPS has the highest cache hit count among these three schemes.

In another experiment, we evaluate the effect of varying the value of P_{inter} while having P_{intra} fixed to 10 percent, and the corresponding results are shown in Fig. 8. It can be seen from Fig. 8 that the cache hit ratio of FCB increases prominently as P_{inter} increases. Since there are more intertransaction pages in the coordinator buffer, FCB has a higher cache hit ratio than others. The distributions of intratransaction and intertransaction cache hit counts are shown in Fig. 9. Similarly to that in Fig. 7, the cache hit counts of intertransaction pages for these three schemes increase as P_{inter} increases. FCB emerges as the best method in this case. Note that the cache hit counts of intratransaction pages for FLB are about 0.7 and, thus, negligible in Fig. 9. The reason of having the lowest intratransaction cache hit counts for FLB is mainly due to frequent page replacements in the local buffers.

Based on the forgoing, it is important to note that FPS performs best for transactions with prominent intratransaction locality. On the other hand, FCB outperforms others for transactions with prominent intertransaction locality. From our experimental results, it is predictable that the difference

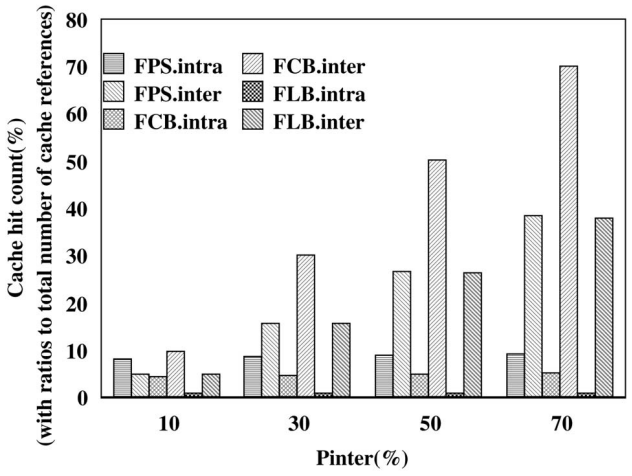

Fig. 9. The distributions of cache hit counts for intratransaction and intertransaction pages with P_{inter} varied.

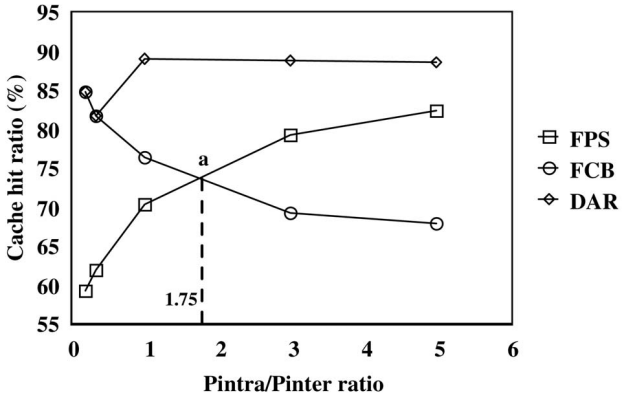

Fig. 10. The cache hit ratios of FPS, FCB, and DAR with $P_{pseudo} = 20$ percent, the coordinator buffer size = 50 percent DBSIZE and P_{intra}/P_{inter} varied.

of cache hit ratios among FCB, FPS, and FLB is negligible when the sum of P_{inter} and P_{intra} is less than 10 percent. Thus, based on our experiments, the temporal locality threshold (i.e., $\frac{0.5}{P_{CB}}$) can be empirically determined to be 0.1.

4.2.2 Performance of DAR

In this section, we set the pseudo percentage of pages, P_{pseudo}, to be 20 percent and then conduct a sensitivity analysis on the ratio of P_{intra} to P_{inter}, i.e., P_{intra}/P_{inter}. Also, without loss of generality, we tentatively set the value of φ to be one for scheme DAR. The selection for the optimal value of φ will be discussed in Section 4.3. The cache hit ratios for FPS, FCB, and DAR with different values of P_{intra}/P_{inter} are shown in Fig. 10.

Note that transactions corresponding to the left-hand side of Fig. 10 have more intertransaction locality, whereas those corresponding to the right-hand side of Fig. 10 possess more intratransaction locality. Therefore, the cache hit ratio of FCB decreases as the value of P_{intra}/P_{inter} increases. In contrast, FPS has more cache hits as the value of P_{intra}/P_{inter} increases. Also, we note that DAR has the highest cache hit ratio, showing the very advantage of employing cache retrieval methods dynamically according to the transaction properties. Fig. 11 shows the distributions of intratransaction and intertransaction cache hit ratios of DAR. It is important to note that, for transactions with more intertransaction (respectively, intratransaction) pages, DAR has higher cache hit ratio of intertransaction (respectively, intratransaction) pages, showing the very adaptability that DAR possesses. This very advantage of DAR enables DAR to perform far better than FPS and FCB, rather than just performs the same as the better of these two schemes.

4.3 Sensitivity Analysis on the Threshold Ratio for DAR

In this subsection, a systematic procedure for determining the optimal operating points of DAR is devised. Specifically, we shall discuss the selection of the threshold ratio of P_{intra}/P_{inter} (i.e., φ), which, as described in Section 3.3.2, is used as a decisive parameter for execution of DAR. The value of φ can be empirically determined by the execution of FPS and FCB with different values of P_{intra}/P_{inter}. More explicitly, the values of φ are determined as the intersection

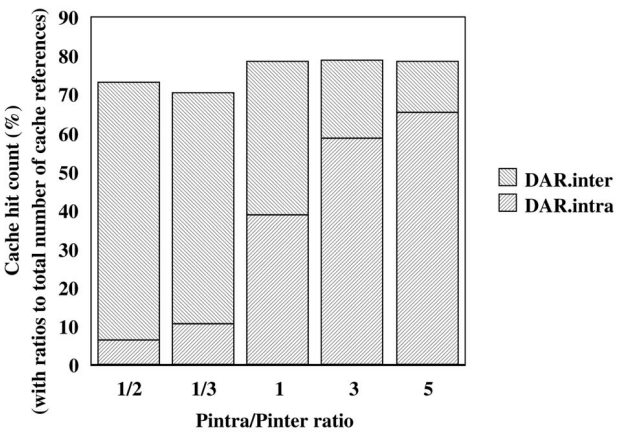

Fig. 11. The distribution of intratransaction and intertransaction cache hit counts of DAR with $\mathrm{P}_{intra}/\mathrm{P}_{inter}$ varied.

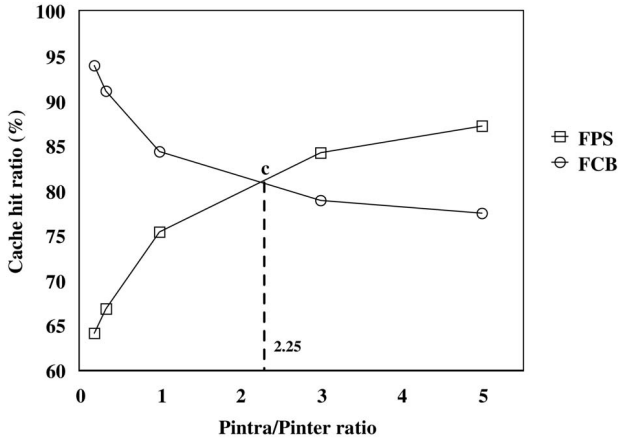

Fig. 13. The value of φ with the coordinator buffer size = 80 percent DBSIZE and $\mathrm{P}_{pseudo} = 20$ percent.

points of FPS and FCB curves in Fig. 10 (i.e., point a where $\varphi = 1.75$), in Fig. 12 (i.e., point b where $\varphi = 1.6$) and, in Fig. 13 (point c where $\varphi = 2.25$). It can be seen that the value of φ varies from 1.6 in Fig. 11 (with the coordinator buffer size being 20 percent of DBSIZE) to 1.75 in Fig. 10 (with the coordinator buffer size being 50 percent of DBSIZE), and then to 2.25 in Fig. 13 (with the coordinator buffer size being 80 percent of DBSIZE). It is noted that as the coordinator buffer size increases, the number cache replacement decreases, in turn leading to a larger cache hit ratio of FCB.

For better clarity, a complete spectrum for the impact of the coordinator buffer size to the value of φ is shown in Fig. 14, where line 1 is for the case when P_{pseudo} is fixed to 20 percent and line 2 is for the case when P_{pseudo} is fixed to 40 percent. Note that the left-upper half of Fig. 14 corresponds to the operating region where scheme FPS is used. In contrast, the right-lower half of Fig. 14 corresponds to the operating region where scheme FCB is used. It can be verified from Fig. 14 that points A, B, and C , respectively, correspond to point a in Fig. 10, point b in Fig. 12, and point c in Fig. 13. From Fig. 14, it can be seen that the value of φ tends to increase when the coordinator buffer size increases, meaning with a larger size of the coordinator buffer, one will favor the use of FCB, which agrees with our intuition.

Note that, it is in general very difficult to guarantee the accuracy of P_{intra} and P_{inter}, whose measurement is usually done empirically [5] and is beyond the scope of this paper. Despite that inaccurate values of P_{intra} and P_{inter} will affect the intersected points of FPS and FCB curves, such an effect is limited to those points very close to thresholds and is likely to diminish soon since the values of P_{intra} and P_{inter} are evaluated periodically. This indicates good tolerance of DAR for the inaccurate values of P_{intra} and P_{inter}.

Furthermore, in order to show the effectiveness of DAR when the percentage of pseudo pages varies, we set the pseudo percentage to 40 percent, and examine the performance of DAR. The cache hit ratios of FPS, FCB, and DAR with $\mathrm{P}_{pseudo} = 40$ percent are shown in Fig. 15. It is seen that the cache hit ratio of DAR with $\mathrm{P}_{pseudo} = 40$ percent is less than that of DAR with $\mathrm{P}_{pseudo} = 20$ percent due to a weaker temporal locality. Note that the experiment in Fig. 10 is the same in Fig. 15 in that both have their coordinator buffer sizes be 50 percent of DBSIZE, but is different from the latter in their values of P_{pseudo} (i.e., $\mathrm{P}_{pseudo} = 20$ percent for Fig. 10 and $\mathrm{P}_{pseudo} = 40$ percent for Fig. 15). It is also noted that with weaker temporal locality for the experiment in Fig. 15, FCB suffers more cache hit loss than FPS, thus causing the intersection point to slightly move to the left.

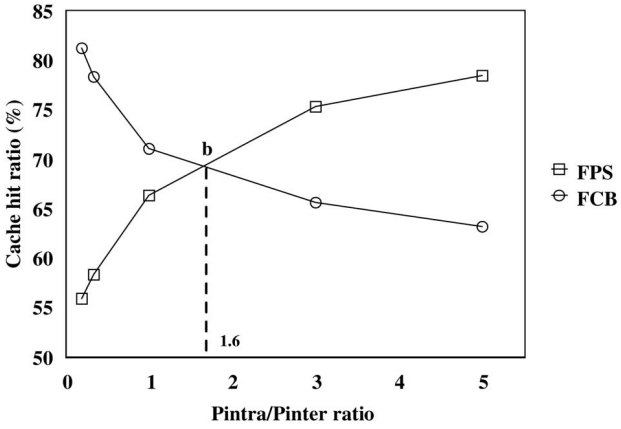

Fig. 12. The value of φ with the coordinator buffer size = 20 percent DBSIZE and $\mathrm{P}_{pseudo} = 20$ percent.

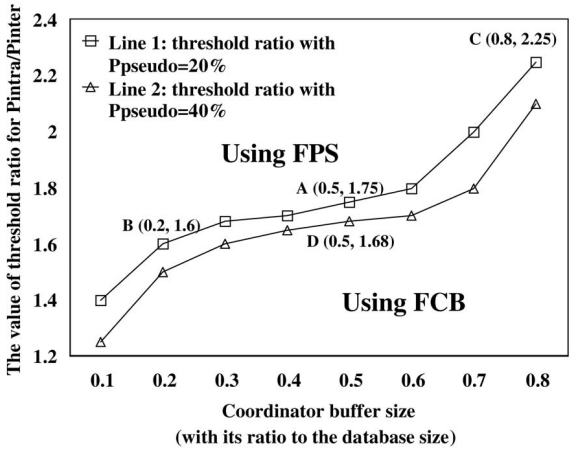

Fig. 14. The value of φ with the coordinator buffer size and P_{pseudo} varied.

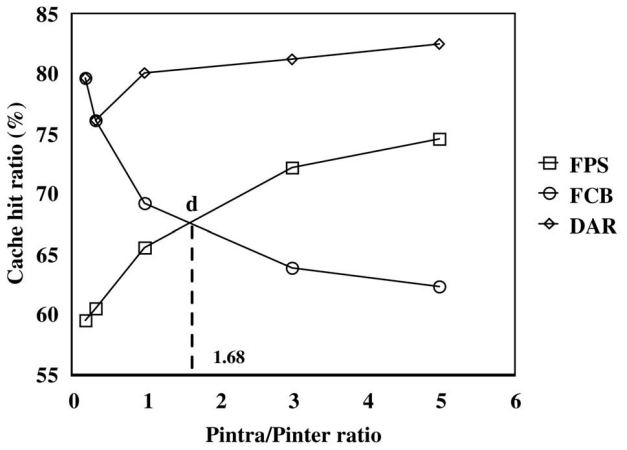

Fig. 15. The cache hit ratios of FPS, FCB, and DAR with $P_{pseudo} = 40$ percent, the coordintor buffer size = 50 percent DBSIZE and P_{intra}/P_{inter} varied.

This accounts for the reason that the resulting value of φ in Fig. 15 (i.e., 1.68, corresponding to point d) is less than that in Fig. 10 (i.e., 1.75, corresponding to point a). A complete spectrum for the impact of the coordinator buffer size to the value of φ with $P_{pseudo} = 40$ percent is shown by line 2 in Fig. 14. It is worth mentioning for transactions with more temporal locality, the advantage of using DAR will be more prominent.

5 Conclusions

We examined in this paper several cache retrieval schemes to improve the efficiency of cache retrieval when a mobile server encounters the service handoff. In particular, we analyzed the impact of using a coordinator buffer to improve the overall performance of cache retrieval. In light of the temporal locality of transactions, we devised a Dynamic Adaptive cache Retrieval scheme (DAR) that can adopt proper cache methods based on some specific criteria devised to deal with the service handoff situation in a mobile computing environment. The performance of these cache retrieval schemes was analyzed and a system simulator was developed to validate our results. It was shown by our results that FPS performed well for transactions with prominent intratransaction locality whereas FCB was favored for transactions with prominent intertransaction locality. A systematic procedure for determining the optimal operating points of DAR has been developed. It is important to note that, for transactions with more intertransaction (respectively, intratransaction) pages, DAR has higher cache hit ratio of intertransaction (respectively, intratransaction) pages, showing the very adaptability that DAR possesses. This very advantage of DAR enables DAR to perform far better than FPS and FCB, rather than just performing the same as the better of these two. Due to its adaptability, DAR is deemed to be particularly effective for a mobile computing environment. It is envisioned that cache retrieval methods for location-dependent data and transcoding data in proxies are important issues yet to be fully explored, and their design and development will be matters of our future research.

Acknowledgments

The authors are supported in part by the Ministry of Education project no. 89-E-FA06-2-4, and the National Science Council project no. NSC 91-2213-E-002-034 and NSC 91-2213-E-002-045, Taiwan, Republic of China.

References

[1] C. Aggarwal, J.L. Wolf, and P.-S. Yu, "Caching on the World Wide Web," *IEEE Trans. Knowledge and Data Eng.*, vol. 11, no. 1, pp. 94-107, 1999.

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

[3] B.Y. Chan, A. Si, and H.V. Leong, "Cache Management for Mobile Databases: Design and Evaluation," *Proc. 14th Int'l Conf. Data Eng.*, pp. 54-63, Feb. 1998.

[4] C.-Y. Chang and M.-S. Chen, "Exploring Aggregate Effect with Weighted Transcoding Graphs for Efficient Cache Replacement in Transcoding Proxies," *Proc. 18th Int'l Conf. Data Eng.*, Feb. 2002.

[5] M.-S. Chen, P.S. Yu, and T.-H. Tang, "On Coupling Multiple Systems with a Global Buffer," *IEEE Trans. Knowledge and Data Eng.*, vol. 8, no. 2, pp. 339-344, Apr. 1996.

[6] V. deNittoPerson, V. Grassi, and A. Morlupi, "Modeling and Evaluation of Prefetching Policies for Context-Aware Information Services," *Proc. Fourth ACM Int'l Conf. Mobile Computing and Networking*, pp. 55-65, Oct. 1998.

[7] 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.

[8] I. Elsen, F. Hartung, U. Horn, M. Kampmann, and L. Peters, "Streaming Technology in 3G Mobile Communication Systems," *Computer*, vol. 34, no. 9, pp. 46-52, 2001.

[9] M.J. Franklin, M.J. Carey, and M. Livny, "Transactional Client-Server Cache Consistency: Alternatives and Performance," *ACM Trans. Database System*, vol. 22, no. 3, pp. 315-363, Sept. 1997.

[10] S. Hosseini-Khayat, "On Optimal Replacement of Nonuniform Cache Objects," *IEEE Trans. Computers*, vol. 47, no. 4, pp. 445-457, 2000.

[11] Q. Hu and D.L. Lee, "Adaptive Cache Invalidation Methods in Mobile Environments," *Proc. Sixth IEEE Int'l Symp. High Performance Distributed Computing*, pp. 264-273, Aug. 1997.

[12] T. Imielinski and B.R. Badrinath, "Wireless Graffiti—Data, Data Everywhere Matters," *Proc. 28th Int'l Conf. Very Large Data Bases*, 2002.

[13] Intelligent Transportation Systems, http://www.artimis.org/stream.php, 2003.

[14] J. Jing, A.K. Elmagarmid, A. Helal, and R. Alonso, "Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments," *ACM J. Mobile Networks and Application*, vol. 2, no. 2, pp. 115-127, 1997.

[15] 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.

[16] A. Kahol, S. Khurana, S.K.S. Gupta, and P.K. Srimani, "A Strategy to Manage Cache Consistency in a Distributed Disconnected Wireless Environment," *IEEE Trans. Parallel and Distributed System*, vol. 12, no. 7, pp. 686-700, 2001.

[17] 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.

[18] W.-C. Lee and D.-L. Lee, "Signature Caching Techniques for Information Filtering in Mobile Enviroments," *ACM J. Wireless Networks*, vol. 5, no. 1, pp. 57-67, Jan. 1999.

[19] Y.-B. Lin, "Modeling Techniques for Large-Scale PCS Networks," *IEEE Comm. Magazine*, vol. 35, no. 2, pp. 102-107, Feb. 1997.

[20] Mobile Streaming in Nokia, http://www.nokia.com/nokia/0,5184,401,00.html, 2003.

[21] C. Mohan, "Caching Technologies for Web Applications," *Tutorial of the 27th Int'l Conf. Vary Large Data Bases*, Aug. 2001.

[22] E. Pitoura and B. Bhargava, "Data Consistency in Intermittently Connected Distributed Systems," *IEEE Trans. Knowledge and Data Eng.*, vol. 11, no. 6, pp. 896-915, Nov./Dec. 1999.

[23] 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.
[24] S. Shekhar, A. Fetterer, and D. Lui, "Genesis: An Approach to Data Disemination in Advanced Travel Information Systems," *IEEE Data Eng. Bulletin,* vol. 19, no. 3, pp. 37-45, 1996.
[25] J. Shim, P. Scheuermann, and R. Vingralek, "Proxy Cache Algorithms: Design, Implementation, and Performance," *IEEE Trans. Knowledge and Data Eng.,* vol. 11, no. 4, pp. 549-561, 1999.
[26] A. Silverschatz and P.B. Galv, *Operating System Concepts.* Addison-Wesley, 1994.
[27] "Streaming in HP,"http://www.hpl.hp.com/research/mmsl/projects/streaming.htm, 2003.
[28] C.-J. Su and L. Tassiulas, "Joint Broadcast Scheduling and User's Cache Management for Efficient Information Delivery," *Proc. Fourth ACM/IEEE Int'l Conf. Mobile Computing and Networking,* pp. 33-42, Oct. 1998.
[29] U. Varshney and R. Vetter, "Emerging Mobile and Wireless Networks," *Comm. the ACM,* vol. 43, no. 6, pp. 73-81, June 2000.
[30] A. Wolski, "Database Replication for the Mobile Era," *Tutorial of the 18th Int'l Conf. Data Eng.,* Feb. 2002.
[31] K.-L. Wu, P.-S. Yu, and M.-S. Chen, "Efficient Caching for Wireless Mobile Computing," *J. Distributed and Parallel Databases,* vol. 6, no. 4, pp. 351-372, 1998.
[32] J. Xu, Q. Hu, D.-L. Lee, and W.-C. Lee, "SAIU: An Efficient Cache Replacement Policy for Wireless On-Demand Broadcasts," *Proc. ACM Ninth Int'l Conf. Information and Knowledge Management,* pp. 46-53, 2000.
[33] J. Xu, X. Tang, and D. Lee, "Performance Analysis of Location-Dependent Cache Invalidation Schemes for Mobile Environments," *IEEE Trans. Knowledge and Data Eng.,* vol. 15, no. 2, pp. 474-488, Mar./Apr. 2003.
[34] T. Yoshimura, Y. Yonemoto, T. Ohya, M. Etoh, and S. Wee, "Mobile Streaming Media Content Delivery Network Enabled by Dynamic SMIL," *Proc. 11th Int'l World Wide Web Conf.,* pp. 651-661, May 2002.

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 National Taiwan University, Taiwan, ROC in 2001. Dr. Peng is currently an assistant professor in the department of computer science and information engineering at the National Chiao Tung University. His research interests include mobile computing, mobile data management, and data mining. He is a member of the Phi Tau Phi scholastic honor society and the IEEE.

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 the chairman of the Graduate Institute of Communication Engineering and also a professor in both the Electrical Engineering Department and Computer Science and Information Engineering Department of the National Taiwan University, Taipei, Taiwan. He was a research staff member at IBM Thomas J. Watson Research Center, Yorktown Heights, New York, from 1988 to 1996. His research interests include database systems, data mining, mobile computing systems, and multimedia networking, and he has published more than 170 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* on data mining and parallel database areas from 1997 to 2001, is on the editorial board of the *VLDB Journal*, *International Journal of Knowledge and Information System (KAIS)*, *Journal of Information Science and Engineering (JISE)*, and *Journal of the Chinese Institute of Electrical Engineering*, was a distinguished visitor of IEEE Computer Society for Asia-Pacific from 1998 to 2000, and program chair of PAKDD-02 (Pacific Area Knowledge Discovery and Data Mining), program vice-chairs of VLDB-2002 (Very Large Data Bases) and ICPP 2003, general chair of Real-Time Multimedia System Workshop in 2001, program chair of IEEE ICDCS Workshop on Knowledge Discovery and Data Mining in the World Wide Web in 2000, and program cochair of the International Conference on Mobile Data Management (MDM) in 2003, the International Computer Symposium (ICS) on Computer Networks, Internet and Multimedia in 1998 and 2000, and ICS on Databases and Software Engineering in 2002. 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 at the 11th IEEE International Conference on Data Engineering in 1995 and also a guest coeditor for the *IEEE Transactions on Knowledge and Data Engineering* special issue for data mining in December 1996. He holds, or has applied for, 18 U.S. 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 in Taiwan and the Outstanding Innovation Award from IBM Corporate for his contribution to a major database product, and also received numerous awards for his research, teaching, inventions, and patent applications. He coauthored with his students for their works which received ACM SIGMOD Research Student Award and Long-Term Thesis Awards. Dr. Chen is a Fellow of IEEE and a member of ACM.