

A Dynamic and Adaptive Cache Retrieval Scheme for Mobile Computing Systems

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

Recent advances in wireless technologies have made the mobile computing a reality. In order to provide services of good quality to mobile users and improve the system performance, the mobile computing system is usually of a distributed server architecture. As users move to a new service area, the new server is expected to take over the execution of running programs for mobile users so as to reduce the communication overhead of the mobile system. This procedure is referred to as service handoff. Note that when service handoff occurs, the cache of the new server does not contain any data entry that was accessed by prior transactions and the new server will thus lose its advantages for cache access. To remedy this, we examine 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 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 is analyzed and a system simulator is developed to validate our results.

1. Introduction

Due to recent technology advances, an increasing number of users are accessing various information systems via wireless communication. Such information systems [1, 2, 6, 16] as mobile shopping, banking, wireless conferencing, are being provided by Information Service and Application Provider (ISAP) [8], and the mobile users are able to access such information via

wireless communication from anywhere at any time.

The ISAP maintains a set of servers which contain information and applications. For cost-performance reasons, the ISAP usually adopts a distributed server architecture [8, 10, 11, 13], in which a service area refers to the converge area where the server can provide service to mobile users. When mobile users enter a new service area, the new server is expected to accept and continue the running applications seamlessly so as to reduce the communication cost of the mobile computing system. This procedure is called *service handoff* [8, 10, 11, 13, 14]. 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 [8, 11, 13, 14].

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 prior transactions. The new server thus loses its advantages for cache access after service handoff. In order to maintain the use of cache after service handoff, the new server should employ proper cache schemes that are suitable for such a mobile computing environment. This is the very problem we shall address in this paper.

Generally speaking, three caching scenarios are considered in this study. The first one is to access cache data in the server (to be referred to as FLB, standing for “from local buffer”). Since the server processes the transactions from mobile users, the buffer of the server contains recently used data. 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 allowing a

coordinator to have a coordinator buffer keeping data to be shared by all servers [4, 5, 7, 9]. It has been reported that a coordinator buffer is useful in improving the system performance and scalability [5]. The third scheme is to access cache data from the prior server of running transactions (to be referred to as FPS, standing for “from the previous server cache”). Clearly, this scheme is included for our evaluation due to the very nature of mobile computing.

Figure 1 illustrates a scenario for cache retrieval when service handoff occurs. In the beginning, served by server A, mobile user 1 submits a transaction to server A. According to the transaction properties that will be discussed in detail later, server A will use either FCB or FLB to pre-cache some data in its cache when the transaction starts to process. Suppose that server A uses FCB to pre-cache data and mobile user 1 moves to a new service area which is covered by server B. The running applications of mobile user 1 will be transferred to server B for execution. Then, server B may use one of the three schemes: FPS (i.e., server A), FCB (i.e., coordinator buffer) and FLB (i.e., from its own local buffer) for cache access. 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 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 is the object of this study.

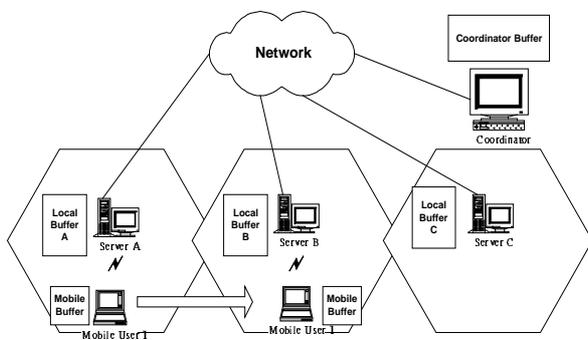


Figure 1. A cache retrieval problem in mobile computing system.

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 will be built to validate our

results. It is shown that temporal locality of transactions (which refers to the feature that consecutive transactions are likely to access same data [4, 5, 9]) plays an important role for cache access in mobile computing and also that adopting cache methods dynamically, as DAR does, is very important and will lead to prominent system improvement.

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. We mention in passing that the authors of [3] proposed some cache invalidation strategies and addressed the impact of disconnection times of clients to the overall performance. In [17], the authors proposed an energy-efficient cache invalidation scheme in which the cache invalidation will take place if the cache data objects are frequently updated ones. The attention of these studies [3, 12, 17] was mainly paid to the cache invalidation scheme for the cache in a mobile unit. In addition, the use of a coordinator in the mobile computing system was pointed out in [8, 13, 14, 15] 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.

This paper is organized as follows. The three cache retrieval methods (FLB, FCB and FPS) are examined in Section 2. The DAR scheme and corresponding decision rules are developed in Section 3. Simulation results are presented and analyzed in Section 4. This paper concludes with Section 5.

2. Cache Retrieval Methods

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. For applications with temporal locality, the data pages which these transactions access can be further divided into two types of pages, namely, pages with intra-transaction locality and pages with inter-transaction locality [9]. Intra-transaction 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, inter-transaction locality refers to the feature that the same data pages are usually shared by some consecutive transactions.

2.1 Description of Three Caching Methods

We now describe three caching methods. As can be seen later, depending on the transaction properties, FLB, FPS and FCB have their own advantages.

FLB (Caching from Local Buffer)

Clearly, since the server has its own local buffer, it could get cache data from its local buffer directly. In general, the server will fetch cache data due to cache miss in the beginning of transaction execution. This is usually called cache warm-up.

FPS (Caching from the Previous Server)

For a transaction with higher intra-transaction locality and lower inter-transaction locality, getting the cache data from the previous server is useful and effective for mobile computing. Figure 2 illustrates such a scenario. Server SA contains the cache pages 44, 26 and 17, and the coordinator contains the cache pages 44, 39 and 40 after server SC writes its cache buffer. These transactions also share one page 44. Under the assumption that the characteristics of workload are of high temporal locality and with few common pages, the mobile user requests the cache pages 44, 26 and 18. As such, when service handoff occurs, getting the cache data from the previous server will be more effective than other schemes.

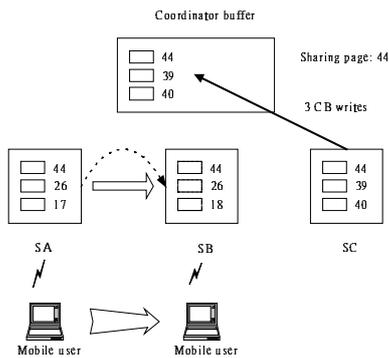


Figure 2. Scenario of getting cache data from previous server.

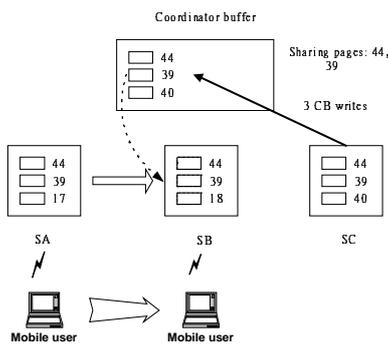


Figure 3. Scenario of getting cache data from coordinator.

FCB (Caching from Coordinator Buffer)

If the transaction property is update-intensive and

transactions possess higher inter-transaction locality, getting cache data from the coordinator buffer will be cost-effective. In Figure 3, the sharing pages are 44 and 39. Assume that the mobile user is using the data pages 44, 39 and 18. When server SB which the mobile user is with gets the cache data from the coordinator, server SB will have the most recent pages 44 and 39, and only incur one cache miss for page 18 and one cache replacement for page 40. 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. Figure 4 shows the general cache retrieval scheme of our 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.

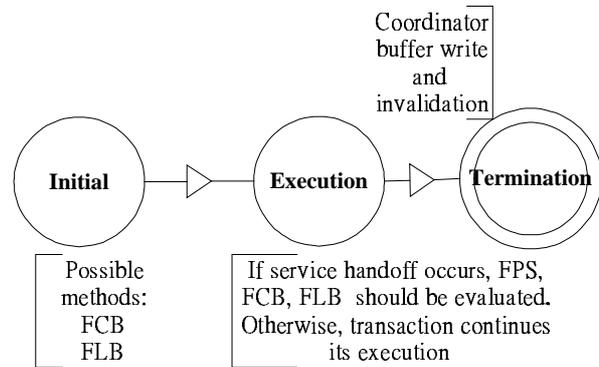


Figure 4. A general cache retrieval scheme called Dynamic and Adaptive cache Retrieval scheme (DAR).

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, will be considered. Note that since the transaction just started, FPS is not proper for this initial phase. These two methods 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 as well as activate the cache invalidation scheme to invalidate other caches in this mobile system. 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 examined 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 the example scenario in Figure 5 where a short transaction is executed by a mobile computer. The transactions are update-intensive and have many sharing pages. The transaction properties are taken into consideration to decide which scheme to use. Qualitatively speaking, since the transaction is update intensive and has inter-transaction locality, FCB tends to perform better.

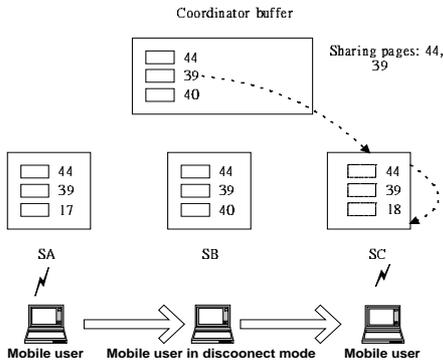


Figure 5. Scenario for short-transaction and disconnection mode of mobile user

Note that due to the inter-transaction locality, the buffer in the coordinator maintains many sharing pages and the server is thus very likely able to get pages from the coordinator buffer. On the other hand, when inter-transaction locality is absent, FLB tends to perform better than FCB because that cache retrieved from FCB will

incur cache replacement, which does not happen when FLB is used.

3.2 Caching Schemes for the Execution Phase

We now consider caching schemes for a transaction in the execution phase. With the transaction execution time being long enough, the transaction processing will 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.

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.

Intra-transaction page probability represents the percentage of pages that demonstrate the intra-transaction locality, whereas inter-transaction page probability represents the percentage of pages that demonstrate the inter-transaction locality. The attributes of inter-transaction page, such as read or update, depend on the update probability for inter-transaction pages. Each transaction is assumed to process an average of T pages. Also, the size of cache in the server buffer is S . The cache miss and the cache replacement cost of each page are denoted by CM and CR , respectively. A description of symbols is given in Table 1.

Table 1. Description of symbols

Description	Symbol
Number of pages that a transaction accesses	T
Intra-transaction page probability	α
Inter-transaction page probability	β
Update probability for inter-transaction page	γ
Intra/Inter ratio for DAR decision in the execution phase	φ
Number of pages available in the server	S
Cache miss cost of one page	CM
Cache replacement cost of one page	CR

3.3.1 Decision Rule for the Initial phase

The number of inter-transaction pages among transactions can be expressed as $T * \beta$. Then we consider the cache miss and cache replacement cost between FCB and FLB. Clearly, the FLB has $CM * T$ cache miss cost.

On the other hand, using FCB to retrieve cache, the cache contains $T\beta$ inter-transaction pages and some other pages which are not available in the cache. Clearly, accessing these pages incurs cache miss and replacement cost. This cost can be expressed as $(CM+CR)(T-T\beta)$. To facilitate our presentation, we denote the minimal number of inter-transaction pages as ε , and use ε as a threshold to determine whether FCB or FLB should be used. Formally, ε is determined from following formula:

$$(CM+CR)(T-\varepsilon) < CM*T$$

$$\Rightarrow \varepsilon > \frac{CR}{(CM+CR)}*T$$

The minimal number of inter-transaction pages indicates that if the $T\beta$ is larger than ε , one should use FCB. Otherwise FLB should be used. In brief, the decision rule is as follows.

Decision Rule for the Initial Phase:

Determine the minimal number of inter-transaction pages ε from
 $(CM+CR)(T-\varepsilon) < CM*T$
 If $(T\beta > \varepsilon)$ then
 Using FCB
 else
 Using FLB.

3.3.2 Decision Rules for the Execution phase

According to the transaction properties, we evaluate the FPS and FCB to decide which one to employ. Specifically, the number of intra-transaction pages in a transaction is αT and the number of inter-transaction pages in a transaction is $T\beta$. We denote the minimal number of pages with temporal locality as σ . In general, if $\alpha T + T\beta$ is less than σ , meaning that temporal locality is not prominent, FLB is used. However, if the transaction property has prominent temporal locality (i.e., $\alpha T + T\beta$ is larger than σ), we shall select the schemes from FCB or FPS. Specifically, use FPS if $\alpha/\beta > \varphi$ (meaning that intra-transaction locality is the major temporal locality), and use FCB if $\alpha/\beta < \varphi$ (meaning that inter-transaction locality is the major temporal locality).

Use FPS or FCB to retrieve cache, the cache contains intra-transaction, inter-transaction pages and some other pages which are not available in the cache. Hence, the extra page access incurs cache miss and replacement cost. Therefore, similarly to the derivation in Section 3.3.1, we can determine σ as follows.

$$(T-\sigma)(CM+CR) < CM*T$$

$$\Rightarrow \sigma > \frac{CR}{(CM+CR)}*T$$

Once the corresponding thresholds, i.e., ε , φ and σ , 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.

Decision Rule for the Execution Phase

Determine the minimal number of temporal locality pages σ from
 $(CM+CR)(T-\sigma) < CM*T$
 If $(\alpha T + T\beta) > \sigma$ then
 Using FLB
 else
 $\alpha/\beta > \varphi \Rightarrow$ using FPS
 $\alpha/\beta < \varphi \Rightarrow$ using FCB

4. Performance Study of Cache Retrieval Schemes

In this section, we describe our simulation model Section 4.1. The experimental results of the simulation are then discussed in Section 4.2.

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 the 8x8 mesh topology network. Each node in this 8x8 mesh topology represents one information server. Hence, 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 several operations that are modeled as a uniform distribution between $SITE_OP-2$ and $SITE_OP+2$ to the server. After the server finishes these operations, the mobile user moves to one of the neighboring servers depending on the routing function of the mesh network, with a probability of 0.25. The total number of operations in a transaction is represented as TXN_SIZE . Table 2 summaries the definitions and default values of some primary simulation parameters. The size of the cache in each server is 50% of DB_SZIE . Similarly, the size of the cache in the coordinator server is 50% of DB_SIZE . Objects in the cache are managed according to LRU (Least Recently Used) replacement policy.

It is assumed that there are 2600 objects in the database of each server and three kinds of objects in the database. The first kind of objects is inter-transactions object (100 objects). The access to inter-transaction

objects is modeled by a uniform distribution with an interval of 100. The second one is intra-transaction object (1000 objects).

Table 2. The parameters in this simulation model.

Notation	Definition	Value
Arrival	number of the customers arrive to each server	exponential distribution with mean 10
SITE_OP	number of operations performed in the server	uniform distribution with mean SITE_OP
TXN_SIZE	number of operations in a transaction	60
DB_SIZE	number of pages in the server	2600 objects
Num_cpus	number of processors in the server	4
Num_disks	number of disks in the server	8
Server_cache_size	number of cache entries in the server	50% of DB_SIZE
Coord_cache_size	number of cache entry in the coordinator server	50%, 80%, 100% of DB_SIZE
CPU_delay	CPU time for accessing a page	12 milliseconds
CC_request_delay	CPU time for CC_request	3 milliseconds
IO_delay	I/O time for accessing a page	35 milliseconds
Update_txn_prob	update transaction percentage	0.1
Write_op_prob.	percentage of write operations for an update transaction	0.5
P_{intra}	percentage of intra-transaction pages	various values used
P_{inter}	percentage of inter-transaction pages	various values used
P_{pseudo}	percentage of pseudo pages	various value used

In order to capture the nature of intra-transaction locality, we divide the 1000 objects into $1000/TXN_SIZE$ groups. Each transaction will access intra-transaction objects in its own group. To model the temporal locality of intra-transaction, we use a normal distribution for object selection. The third one is pseudo objects (1500 objects). Similarly, the access to pseudo object is modeled by a uniform distribution with an interval 1500. Overall,

the total objects that are accessed by a transaction consist of 3 parts, i.e., P_{intra} portion intra-transaction objects, P_{inter} portion inter-transaction objects, and $(1-P_{intra}-P_{inter})$ portion pseudo objects.

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 on the cache hit ratio 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 changing the percentage of intra-transaction pages (P_{intra}) that are accessed by transactions. Specifically, we fix the percentage of inter-transaction (P_{inter}) to 10%, and examine which scheme is suitable when the value of P_{intra} increases. Figure 6 shows the cache hit ratios of FPS, FCB and FLB.

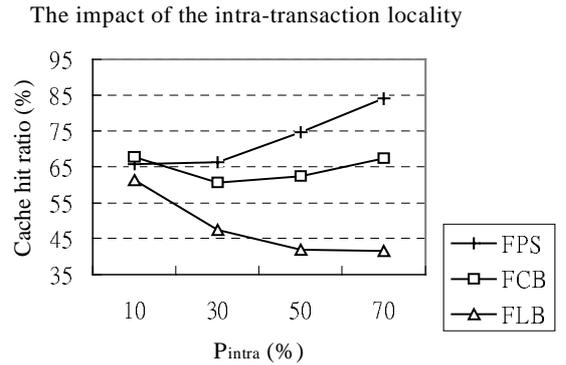


Figure 6. The cache hit ratios of FPS, FCB and FLB by changing P_{intra}

It can be seen from Figure 6 that FLB has the lowest cache hit ratio and the cache hit ratio of FPS tends to increase when P_{intra} becomes larger. 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 cache is much more intensive. From these results, it is seen that FPS performs well for transactions with large value of P_{intra} .

The distributions of intra-transaction and inter-transaction cache hit ratios are showed in Figure 7. It can be seen from Figure 7 that in these three schemes the cache hit ratio for pages with intra-transaction locality increases as P_{intra} increases. This phenomenon is more prominent for the case of FPS. In fact, FPS has the

highest cache hit ratio among these three schemes.

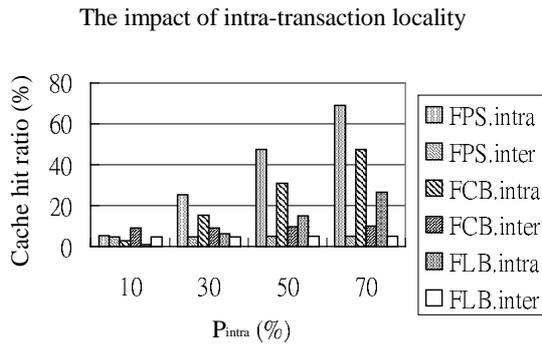


Figure 7. The distributions of intra-transaction and inter-transaction cache hit ratios of FPS, FCB and FLB.

Figure 8 shows the results with various values of inter-transaction percentages when the intra-transaction percentage (P_{intra}) is fixed to 10%. The cache hit ratios for both FPS and FLB decrease as P_{inter} increases. In contrast, the cache hit ratio of FCB tends to increase as P_{inter} increases. Since there are more inter-transaction pages in the coordinator cache, FCB has a higher cache hit ratio than others. The distributions of intra-transaction and inter-transaction cache hit ratios are showed in Figure 9. Similarly to that in Figure 7, the cache hit ratios of inter-transaction for these three schemes increase as P_{inter} increases. FCB emerges as the best solution in this case.

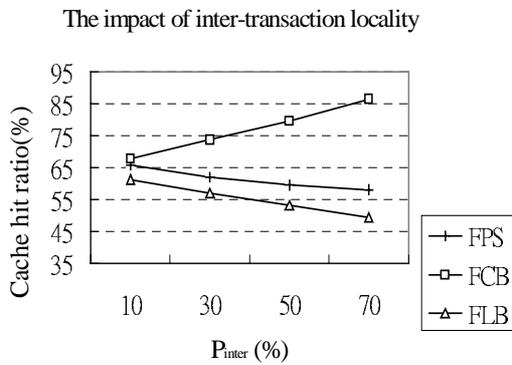


Figure 8. The cache hit ratios of FPS, FCB and FLB by changing P_{inter}

Based on the forgoing, it is important to note that FPS performs best for transactions with prominent intra-transaction locality. On the other hand, FCB outperforms others for transactions with prominent inter-transaction locality.

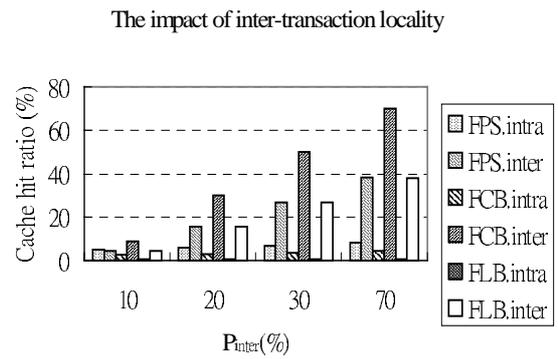


Figure 9. The distributions of intra-transaction and inter-transaction cache hit ratios of FPS, FCB and FLB.

4.2.2 The Performance of DAR

In this section, we set the pseudo percentage of pages, P_{pseudo} , to be 20% and then conduct a sensitivity analysis on the ratio of P_{intra} and P_{inter} , i.e., P_{intra}/P_{inter} . Also, without loss of generality, we tentatively use one in this subsection as the value of φ for DAR. The cache hit ratios for FPS, FCB and DAR with different values of P_{intra}/P_{inter} are shown in Figure 10.

Note that transactions corresponding to the left-hand side of Figure 10 have more inter-transaction locality, whereas those corresponding to the right-hand side of Figure 10 possess more intra-transaction locality. Therefore, the cache hit ratio of FCB decreases as 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.

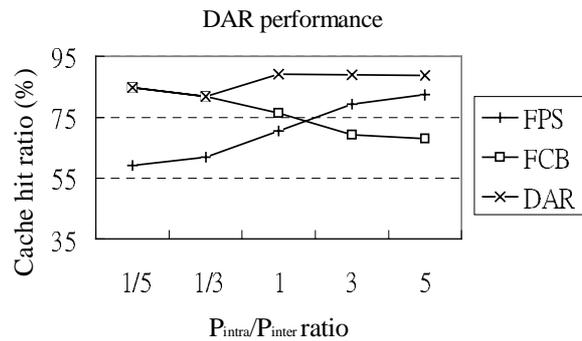


Figure 10. The cache hit ratios of DAR, FPS and FCB with varying P_{intra}/P_{inter} .

5. Conclusions

We examined in this paper several cache retrieval schemes to improve the efficiency of cache retrieval. 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 temporal locality had a significant impact on the performance of cache retrieval methods. Specifically, FPS performed best for transactions with prominent intra-transaction locality whereas FCB outperformed others for transactions with prominent inter-transaction locality. By adaptively adopting the advantages of different cache retrieval methods, DAR performed very well and was particularly effective for a mobile computing environment.

Acknowledgements

The authors are supported in part by the National Science Council, Project No. NSC 87-2213-E-002-009 and NSC 87-2213-E-002-101, Taiwan, Republic of China.

Reference

- [1] R. Alonso and H. Korth. Database system issues in nomadic computing. In *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, pp. 388-392, 1993.
- [2] B. Bruegge and B. Bennington. Applications of Mobile Computing and Communication. *IEEE Personal Communication*, pp. 64-71, Feb. 1996.
- [3] D. Barbara and T. Imielinski. Sleepers and workholics: Caching in mobile distributed environment. In *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, pp. 1-12, 1994.
- [4] M. J. Carey, M. J. Franklin, E. J. Shekita. Data Caching Tradeoff in Client-Server DBMS Architecture, In *Proc. of the ACM SIGMOD*, pp. 357-366, May 1991.
- [5] M.-S. Chen, P.-S. Yu and T.-H. Tang. On Coupling Multiple Systems with A Global Buffer. *IEEE Trans. on Knowledge and Data Engineering* 8(2): 339-344, April 1996.
- [6] N. Davies, G. S. Blair, K. Cheverst and A. Friday. Supporting Collaborative Applications in a Heterogeneous Mobile Environment. *Computer Communication Special Issue on Mobile Computing*, 1996.
- [7] A. Demers, K. Petersen, M. Spreitzer, D. Terry, M. Theimer and B. Welch. The BAYOU Architecture: Support for Data Sharing Among Mobile Users. In *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 2-7, Dec. 1994.
- [8] A. Elmagarmid, J. Jain and T. Furukawa. Wireless Client/Server Computing for Personal Information Services and Applications. *ACM SIGMOD Record* 24(4): 16-21, Dec. 1995.
- [9] M. J. Franklin, M. J. Carey, and M. Livny. Transactional Client-Server Cache Consistency: Alternatives and Performance. *ACM Transaction on Database System* 22(3): 315-363, Sep. 1997.
- [10] T. Imielinski and B. R. Badrinath. Mobile wireless computing. *Communication of ACM* 37(10):18-28, Oct. 1994.
- [11] J. Jing, O. Bukhres and A. Elmagarmid. Distributed Lock Management for Mobile Transactions. In *Proc. of the 15th Int. Conf. on Distributed Computing Systems*, pp. 118-126. June. 1995.
- [12] J. Jing, A. K. Elmagarmid, A. Helal, R. Alonso: Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments. *ACM Mobile Network* 2(2): 115-127, 1997.
- [13] R. Jain and N. Krishnakumar. Network Support for Personal Information Services to PCS Users. *IEEE Conf. Networks for Personal communications*, Mar. 1994.
- [14] N. Krishnakumar and R. Jain. Escrow Techniques for Mobile Sales and Inventory Applications. *ACM Wireless Network* 3(3):235-246, 1997.
- [15] E. Pitoura and B. Bhargava. Revising Transaction Concepts for Mobile Computing. In *Proc. of the first workshop on Mobile Computing Systems and Applications*, pp.: 164-168, 1994.
- [16] M. Satyanarayanan. Mobile Information Access. *IEEE Personal Communication*, pp. 26-33, Feb. 1996.
- [17] K.-L. Wu, P.-S. Yu and M.-S. Chen. Energy-Efficient Caching for Bandwidth-Limited Wireless Mobile Computing. In *Proc. of the 12th IEEE Intern'l Conf. on Data Engineering*, Feb. 26 – Mar. 1, pp. 335-343, 1996.