An Efficient Cache Invalidation Strategy in Mobile Environments

Po-Jen Chuang and Ching-Yueh Hsu
Department of Electrical Engineering, Tamkang University
Tamsui, Taipei Hsien, Taiwan 25137, R.O.C.
E-mail: pjchuang@ee.tku.edu.tw

Abstract

This paper presents a new cache invalidation strategy able to maintain data consistency between the server and mobile clients in an efficient way in mobile communications.

1. Introduction

A mobile environment consists of a large number of mobile clients and a small number of powerful database servers. The servers are connected through a wired network, and the mobile clients are connected to the server through a wireless communication channel. An effective mechanism to reduce data access time and bandwidth utilization in mobile communication is to cache data items at the end of the mobile clients. To avoid data inconsistency, or to maintain data coherence, between items in the server and their corresponding cached items in the mobile clients, the server will broadcast invalidation reports (IRs) to the mobile clients periodically or aperiodically. The mobile clients then invalidate their cached data items following the content of the received IRs. An IR usually contains such information as data IDs or timestamps, but not their values because carrying the values will consume too much bandwidth.

Various cache invalidation strategies [1-6] have been proposed in recent years, including the timestamp (TS) [1], the bit-sequence (BS) [2], the dual-report (DRCI) [3] and the invalidation by absolute validity interval (IAVI) [4] strategies. Some strategies, such as the TS algorithm, verify the validity of the cached data items through the uplink and downlink channels between the server and the mobile clients. When a user needs a certain cached data item that has been invalidated, the mobile client will file a request to the server via the uplink channel and receive results from the server via the downlink channel.

The performance of cache invalidation can be seriously damaged if communication between the client and the mobile clients is disconnected to conserve bandwidth or power utilization. This paper presents an efficient cache invalidation strategy with reduced IR sizes, adapted timestamps and without unnecessarily invalidated data items. Experimental evaluation shows that the new strategy performs favorably -- with reduced data access time and bandwidth utilization – in verifying the validity of cached data items.

2. The proposed strategy

A desirable invalidation strategy should be able to maintain data consistency between the server and the mobile clients using possibly the least resources. This can be achieved by minimizing either the IR sizes or the utilization of link channels. Based on this, our new strategy is devised in such a way that (1) the size of the IRs and the number of involved link channels are both significantly reduced to save bandwidth utilization and data access time, and (2) unnecessary invalidation of cached data items is avoided.

Our new cache invalidation strategy works as follows: If the mobile client is always on the connection mode, the server will broadcast aperiodical IRs to the mobile clients to maintain data consistency between the two parties. Aperiodical broadcasting enables the mobile client to provide a requesting user with the needed and updated information in the earliest convenience. (If the server broadcasts periodical IRs at a certain time interval, users who make cache requests between two broadcasting intervals will not get the requested information unless the mobile clients can verify the validity of the cached items.) If a user sends in queries at the time when the mobile client gets reconnected after T_{lb} (the time when the last IR is received by the mobile client), the mobile client will uplink the T_{lb} value and all query items to the server.

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE

The server then broadcasts the IR consisting only of the IDs of the updated data items since T_{lb} and of the latest updated timestamp T. Thus, the IR in our strategy is reduced to a smaller size.

To be more specific, a user's query will be handled in this way. If the queried items are not in the cache memory, the mobile client will file cache requests to the server. If the queried items are in the cache and their validity is certain: Use the cached data if they are valid; otherwise, file the cache request to the server. If the queried items are in the cache memory but their validity is uncertain (due to disconnection): The mobile client needs to verify the validity of all cached items by invalidation reports. At this point, the mobile client will uplink the T_{lb} value and all query items to the server through the uplink channels. When the server receives T_{lb}, it will detect how many IRs the mobile client lost during the disconnection period. If the mobile client didn't lose any IRs during the disconnection period (i.e., no cached items in the mobile client need to be invalidated), the server will downlink the queried items only. Otherwise, the server will broadcast both the IR and the queried items to the mobile client. The IR now consists only of the IDs of the updated data items since T_{lb} and among these items the latest updated timestamp T. The mobile client then uses these items to answer the user's query, to invalidate the updated data items since T_{lb}, and to set its T_{lb} to T. Thus involving only a little uplink channel bandwidth (T_{lb}), the mobile client obtains IRs he had lost during disconnection, without having to wait for the next periodical IR.

When the server broadcasts an IR to the mobile client who uplinks T_{lb}, the IR will also be received by the other mobile clients in the same area. Then, how do these mobile clients make use of the received IR? As a mobile client keeps T_{lb} even during the disconnection time, when he receives an unrequested IR, he can go ahead compare T_{lb} with T. If T_{lb}=T in the IR, the cached items in the mobile client are still valid. If T_{lb}<T, the cached items are not updated to time and their validity should be checked according to the content of the received IR. The value of T_{lb} will then be set to T.

3. Performance evaluation

Extensive simulation runs have been conducted to evaluate and compare the performance of our cache invalidation strategy with that of the TS, BS, DRCI and IAVI strategies. The simulation model comprises the server, mobile clients, uplink channels and downlink channels. The results are given below. (Details of the simulation model and a number of figure presentations of the results are dropped from the paper due to very limited space.)

The size of each data item in IRs. Among all five strategies, our strategy has the smallest data items in IRs since only the ID is included in the IR for each data item. This can be translated into reduced downlink channel bandwidth utilization.

The number of unnecessarily invalidated data items. The result in Figure 1 indicates no unnecessarily invalidated data items for our strategy because it broadcasts IRs aperiodically and offers a special mechanism using T_{lb}. The design helps save bandwidth resources. For the other invalidation strategies, data items will not be "falsely" invalidated when the disconnection time is under 100 sec because the periodically broadcast IRs can verify all cached items. But, when the disconnection time goes over 1000 sec, these strategies begin to lose some IRs. To make up for the loss (of IRs), they simply invalidate all cached items, including those still valid ones. When users make any new queries, the mobile clients needs to file the cache requests to the server even if the users' requests involve cache items that are still valid but have been "wrongly" invalidated. In such a situation, the caching process becomes redundant and resource-wasting. As observed, there are more unnecessarily invalidated data items at disconnection time 1000 sec than at 10000 sec. This is understandable because when the disconnection time reaches 10000 sec, most cached items will have been "necessarily" invalidated.

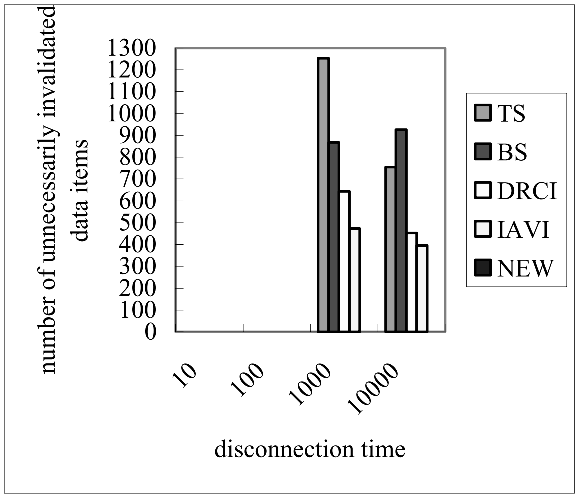

Figure 1. The number of unnecessarily invalidated data items vs. disconnection time.

The number of cache requests filed to the server. The mobile clients need to send cache requests to the server in two situations: (1) when the queried data item is not stored in the cached memory, and (2) when the cached items have been invalidated. The first situation usually happens when the disconnection time is less than 100 seconds. In such a short period of disconnection, the data items which have been stored in the cache still retain their validity, and it is only when a newly queried item is absent from the cache that the mobile client needs to uplink the request to the server. When this happens, every invalidation strategy virtually acts and performs the same: Duly sending the cache request to the server. After the mobile client was disconnected from the server longer than 1000 sec, the other strategies are found having to uplink cache requests more frequently than our strategy. This happens because, during the long disconnection, the other strategies have unnecessarily invalidated data items in the cache, while our strategy generates no unnecessary invalidation at all.

Bandwidth utilization. The downlink channel bandwidth is utilized when the server broadcasts IRs and the requested cache information to the mobile clients. Simulation results show that the BS strategy consumes the biggest amount of downlink bandwidth in such broadcasting because it has the largest IRs. The TS and DRCI strategies, also with large-sized IRs, are the other top downlink channel consumers. By contrast, our proposed strategy utilizes the least amount of downlink channels partly because it has the smallest IRs and partly because it does not excessively invalidate the cache items in the cache after disconnection.

Cache miss ratio. The miss ratio is defined to be the number of inaccessible or invalid data items in the cache over all queried data items. A strategy with low cache miss ratio will be able to handle the users' queries more quickly. It is shown that with longer disconnection time (such as 1000 sec and 10000 sec – the more practical situations), our strategy yields the lowest cache miss ratio among all strategies. This is justifiable because without unnecessary invalidation of the cached items, our strategy will respond to the users' queries more promptly, ensuring higher cache hit ratio (i.e., lower miss ratio) than the other strategies.

Access time. The above cache invalidation strategies are also simulated to obtain their access time vs. a number of parameters (such as the disconnection time, the query arrival time, the update interval, the server database size, and the downlink and uplink channel bandwidth). The access time is the time elapsed from the moment the client submits a request to the point when all the requested data items are downloaded by the client [3]. A perfect server (PS) strategy, in which the server is aware of the contents of all the cached items and broadcasts IRs containing only the updated information of the cached items, is added in the simulation to serve as an optimum. Collected simulation results demonstrate a similar performance trend for all parameters: The proposed new strategy performs better than the other strategies in nearly all cases. Figure 2 depicts the result of access time vs. disconnection time. The disconnection time varies from 10 to 10000 sec; the request arrival interval is 0.5 sec. The result shows that the access time of all strategies remains unaffected when the disconnection time is under 1000 sec (because at this point the strategies can correctly verify their cached items by the broadcast IRs). But when the disconnection time grows up from 1000 to 10000 sec, the access time rises for all strategies. This is because most cached data items are invalidated during the long period of disconnection. As a result, when users send in cache requests, the mobile clients need to file almost all the requests to the server who then responds by broadcasting the needed information, thus slowing down the entire data transmission process.

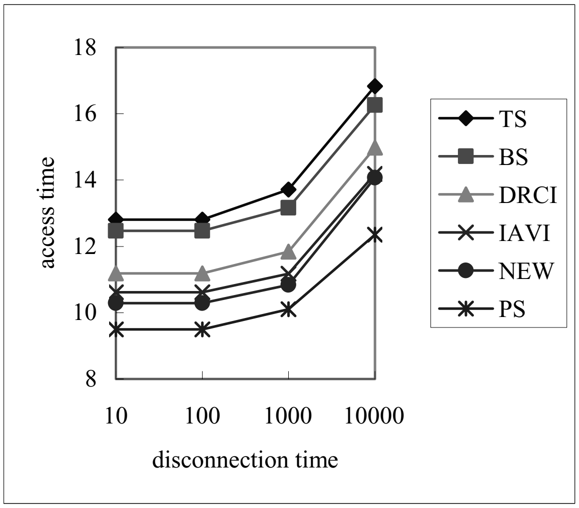

Figure 2. Access time vs. disconnection time.

Energy consumption. The energy consumption contains the energy consumed on invalidating cache items, uplinking requests and downloading the desired data [3]. Thus, energy consumption in our simulation takes into account the following three factors: (1) the size of IRs, (2) the total cost when the mobile clients uplink the T_{lb} value or request data items, and (3) the total amount of data items that the server downlinks to the mobile clients. The energy consumed at receiving 1k-bits of information is defined as 1 unit; transmitting data is assumed to consume 10 times more energy than receiving data, based on [3,7]. As Figure 3 illustrates, energy consumption increases with disconnection time for all strategies. This is because when the disconnection time lengthens, more cached items are invalidated and when users send in cache requests, there will be more data transmission (including increased numbers of IRs) in both the downlink and uplink channels. The figure shows that our strategy consumes less energy than the other four strategies when the disconnection time is below 1000sec. When the disconnection time reaches 1000 sec and beyond, the amount of energy needed by our strategy, though coming closer to that of the IAVI strategy, still remains the lowest.

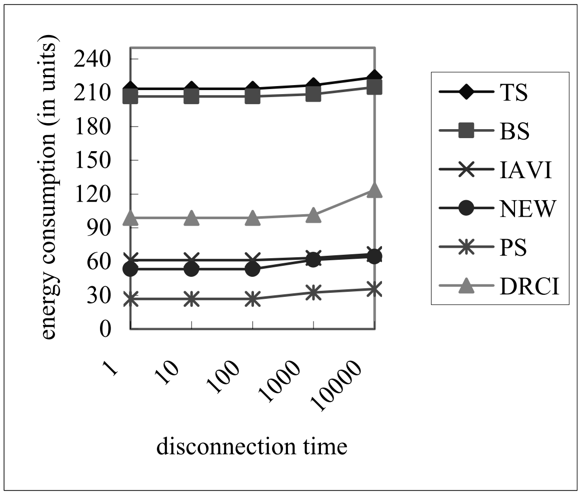

Figure 3. Energy consumption vs. disconnection time.

4. Conclusions

In a mobile environment, caching data items at the mobile clients is important as it reduces the data access time and bandwidth utilization. While caching improves data transmission in mobile communication, maintaining data correctness or consistency between the server and the mobile clients becomes consequential. Cache invalidation is a popular and effective way to maintain such data coherence. In cache invalidation, the server will broadcast data invalidation reports (IRs) to the mobile clients who will then update their cached items according to the reports. This paper presents a new cache invalidation strategy which preserves the advantages of existing strategies, improves on their disadvantages, and turns up its own designs. The center design of our strategy includes reducing the content of IRs, broadcasting IRs aperiodically, providing a special mechanism using T_{lb}, and having no unnecessarily invalidated items. The design makes it possible to maintain data consistency between the server and the mobile clients in a more efficient way, i.e., with less access time and reduced bandwidth consumption. Experimental evaluation proves that our strategy yields better performance than previous strategies in most of the simulation items, such as data access time, energy consumption and so on.

5. Acknowledgments

This work was supported in part by the National Science Council, Taiwan, R. O. C., under Grant No. NSC 91-2213-E-032-013.

6. References

[1] Q. Hu, and D. L. Lee, "Adaptive cache invalidation methods in mobile environments," *Proc. Int'l Conf. on High Performance Distributed Computing,* 1997, pp. 264-273.

[2] J. Jing, A. Elmagarmid, A. Helal, and R. Alonso, "Bit-sequences: an adaptive cache invalidation method in mobile client/server environments," *Mobile Networks and Applications*, Vol. 31, No. 2, pp.115-127, 1997.

[3] K.-L. Tan, J. Cai, and B. C. Ooi, "An evaluation of cache invalidation strategies in wireless environments," *IEEE Tran. on Parallel and Distributed Systems*, Vol. 12, No. 8, pp. 789-807, Aug. 2001.

[4] E. Chan, C.-H. Yuen, K.-Y. Lam, and H.W Leung, "An adaptive AVI-based cache invalidation scheme for mobile computing systems," *Proc. Int'l Conf. on Database and Expert Systems Applications,* 2000.

[5] K.-L. Wu, P. S. Yu, and M.-S. Chen, "Energy-efficient caching for wireless mobile computing," *Proc. Int'l Conf. on Data Engineering*, Feb. 1996, pp. 336-343.

[6] S. H. Nam, I. Y. Chung, and C.-S. Hwang, "An efficient cache invalidation scheme for mobile wireless environments," *Proc. Int'l Conf. on Parallel and Distributed Systems*, 2001, pp.289-296.

[7] G. H. Forman and J. Zahorjan, "The Challenges of Mobile Compting," Computer, Vol. 27, No. 4, pp. 38-47, Apr. 1994.