Journal of Computer Science 7 (3): 374-378, 2011
ISSN 1549-3636

Cache Management for Concurrent Transaction Execution in Mobile Wireless Environment

[1]J.C. Miraclin Joyce Pamila and [2]K. Thanushkodi
[1]Department of Computer Science and Engineering,
Government College of Technology, Coimbatore, India
[2]Director, Academics, Akshaya College of Engineering, Coimbatore, India

Abstract: Problem statement: Improvement in mobile communication technology and increased capability of mobile devices has made transaction processing possible with mobile devices. Transactions access services provided by the data server connected to the packet data network. Direct data access may overload the network and server. Data items are cached to improve data availability. The cache should be managed and maintained effectively to support concurrent transaction execution in mobile wireless environment. This study proposes a cache management strategy that includes cache invalidation and replacement. **Approach:** The contents of the cache are invalidated based on the predicted life time of the data item and the cache contents are replaced based on the degree of sharing. The count of dependent transactions determines the degree of sharing of the data item. **Results:** The experimental results ensure reduced stale hit probability and restart probability of the proposed PLP based cache invalidation technique. The proposed cache replacement policy improves the cache hit ratio at least by 20% when compared to the existing replacement policies. **Conclusion:** The proposed cache invalidation strategy and replacement policy suit well for concurrent transaction execution environment where the degree of data sharing and the dynamism in data update need to be considered.

Key words: Cache invalidation, cache replacement, concurrent transaction execution, mobile transactions, Global System for Mobile Communication (GSM), Universal Mobile Telecommunications System (UMTS), Base Station (BS)

INTRODUCTION

The 2.5G and 3G mobile networks provide improved data services in addition to voice services. Mobile devices are also facilitated with improved computational ability. Public data processing like appointment fixing with doctors, automobile service centres and research labs may be done with Mobile Hosts (MH). When such data services are provided by the Data Servers (DS) of the Packet Data Network (PDN), the mobile transactions may avail them. When the public data items are directly accessed from the server, the response time may increase with increased server and network overhead. So caching of frequently used data items is preferred. Caches are intermediate fast memory which store duplicated data items from the server. But the cached data items must be properly invalidated when the original copy of the data item in the server is updated. Normally invalidation reports are generated when data items are updated in the server, to indicate the status of the updated cached data items. Cache invalidation strategy ensures preservation of data consistency between the cached and original copy. Caches are of limited size. So the cached data items must be effectively replaced when new item is to be cached. The cache replacement policy should ensure increased cache hit ratio.

The General Packet Radio Service (GPRS)/Universal Mobile Telecommunications System (UMTS) based reference architecture is shown in Fig. 1. A static network consists of fixed hosts and Base Transceiver Stations that will interact with MH and wired network. Base transceiver station (Base Station) is a fixed host in the radio subsystem of Global System for Mobile Communication (GSM), to facilitate radio communication with the MH. Base Station (BS) acts as a gateway between wired and wireless networks. It comprises all radio equipments such as transmitter, receiver and signal processing amplifiers needed for radio transmission and reception. Two or more BSs are controlled by a Base Station Controller (BSC). The BSC has a new piece of hardware called a Packet Control Unit (PCU) that directs the data traffic to the GPRS network. UMTS network defines Node B instead of BS of GPRS network with similar functionalities.

Corresponding Author: J.C. Miraclin Joyce Pamila, Department of Computer Science and Engineering, Government College of Technology, Coimbatore, India

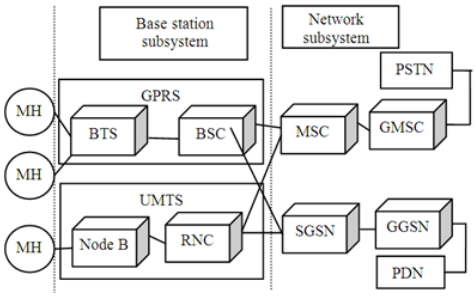

Fig. 1: Reference wireless mobile network architecture

Radio Network Controller (RNC) controls number of Node B elements.

GPRS network adds GPRS support nodes in the network that are responsible for routing and delivery of the data packets between the MH and PDN. To provide packet radio service, GPRS network has Gateway GPRS Support Node (GGSN). It is a networking unit between the GPRS network and external PDN. GGSN that contains routing information for GPRS users performs address conversion and tunnels encapsulated data to user. Serving GPRS Support Node (SGSN), requests user addresses from the GPRS register and keeps track of the individual MH's location. It is responsible for billing information and security functions such as access control. When a packet is transmitted from a PDN to a MH, it is transmitted through GGSN, SGSN and BS.

Related study: In the literature, stateless, stateful and hybrid cache consistency maintenance algorithms for wireless mobile computing environments are proposed. In the stateless approach (Barbara and Imielinski, 1994), the data server is unaware of MHs' cache content and the MH needs to check the validity of the cached data items before using them. Even though stateless approaches employ simple database management at the DS, their scalability and ability to support disconnectedness of the MH are poor. On the other hand, stateful approaches (Kahol *et al*., 2001) are scalable, but have significant overhead due to server database management. In the hybrid approach (Wang *et al*., 2004) the server keeps very limited information about clients' cache status. It achieves good tradeoff between scalability and efficiency, as computation overhead involved is less for database management. Scalable Asynchronous Cache Consistency Scheme (SACCS) (Wang *et al*., 2004) is proposed to maintain MH's cache consistency. SACCS requires the MSS to identify which data objects in its database might be valid in MH's caches. This makes the management of the MSS database much simpler. SACCS does not periodically broadcast IR reducing the frequency of IR through the downlink broadcast channel. Shi et *al*. (2005) proposed semantic cache model and invalidation scheme that reduces IR which also assumes storage of cache in mobile hosts. Moiz and Nizamuddin (2008) proposed a cache invalidation scheme wherein which MH can verify the validity of a cached item by comparing the last update time and its Absolute Validity Interval (AVI). AVI of the data item is set equal to the previous update interval. A cached item is invalidated if the current time is greater than the last update time by its AVI. From the literature survey, it is learnt that in all existing schemes, the MH is assumed to have cache loaded in it. The frequency, with which IR should be generated and sent to the MH, differs based on different schemes. The major constraint in existing cache invalidation schemes is to deal with disconnected MH when IR is sent.

Traditional cache replacement strategies such as Least Recently Used (LRU), Least Frequently Used (LFU) and First In First Out (FIFO) do not consider size of the objects to be cached, as they assume elements of equal size. Generalized target driven replacement policy (Yin *et al*., 2005) under strong consistency model formulated functions for specific targets. A scheme based on data profit (Chand *et al*., 2006) ensured highly associated data items are retained in the cache. A cache replacement policy for location dependent information services (Hiary *et al*., 2009) was proposed using virtual table method. Elfaki *et al.* (2011) proposed cache replacement with LRU and TTL which increased cache hit ratio. All the existing replacement techniques provide solutions for cached data items that may not be shared. The proposed policy retains highly shared data items in the cache.

MATERIALS AND METHODS

Transactions are run in MHs and are called mobile transactions. Transactions that access the shared data item are called as affiliated transactions. Cache is stored in the BS (Miraclin, 2009) of the mobile data network. In all existing caching strategies, the cache is loaded in the MH, which allows the affiliated mobile transactions to manipulate the shared data item independently. It may lead to increased transaction 'restart' rate when the shared data item is updated. In the proposed approach, as the cache is loaded in the BS, independent execution in the MHs is coordinated and 'restart' rates are

reduced. Cache Manager (CM), a software upgrade, is loaded in the BS manages cache with consistency preservation scheme and replacement policy.

Caching strategy: The CM module manages all cache operations and access to the cached data. The CM initially fetches data item from the DS and stores it in the cache. All subsequent requests for the cached data item will be serviced by the CM itself.

The CM stores the data fetched from the DS in the cache along with control information. The structure of the data object in the cache is represented as < D_{id}, V_{id}, PLP, ATC, UC, LUT, LUCLK, DTL> where D_{id} is the identity of the data item, V_{id} is the value of the data item, PLP is Predicted Life Period of the data item, ATC is the Affiliated Transaction Count, UC is a flag to confirm the previous update made, LUT is the transaction that updated the data item lastly, LUCLK is the logical clock to represent the last update timestamp and DTL is a Dependent Transaction List.

When more number of transactions share the data item, they are said to be affiliated transactions and the number of affiliated transactions that access that data item is identified with ATC. The flag UC is set, when the update operation on the data item is confirmed by the DS. DTL keeps track of the dependent transactions' identities. Initially DTL is set to NULL. When affiliated transactions access the shared data item, the dependency between the transactions is identified. When a transaction issues RDR (T_i, D_{id}), it becomes dependent transaction of another transaction T_j, if the data item in the cache is updated earlier by T_j and not yet committed in the DS. DTL is reset to NULL when UC flag of the data item is set to indicate update confirmation the DS. Initially when the CM retrieves the data item from the DS, it sets LUCLK to the time instance at which data is retrieved and ATC of the data item to zero; when subsequent data access requests are made, ATC is incremented. The data in the cache is valid as long as it is not updated in the DS. Lamport's logical timestamp is used to logically represent the timestamp of transaction operations rather than physical timestamps. Lamport's logical clock is a monotonically increasing software counter, whose value need not be physical clock dependent.

Proposed cache invalidation technique: When concurrent transactions issue UDR, the replicated copy in the cache is updated locally. The CM generates the invalidation indication to indicate the update of the data item and forwards it to all connected BSs that have caches stored in it. The broadcast of invalid indication avoids transaction processing with invalid stale data. Invalidation indication has the updated value piggybacked on it, to allow transactions to continue with execution. When the connected BSs that have CMs receive the invalidation indication, they update the data item locally with UC of the data item set to 0. The cache manager also broadcasts the invalid indication to all dependent transactions, only if the ATC ratio is high (>0.6). ATC ratio is the ratio of the affiliated transactions to the total transactions. High ATC ratio indicates that almost all transactions are affiliated transactions.

Simultaneously the threaded CM forwards the update request made by the MH to the DS. The DS after validation does the update and broadcasts the invalidation confirmation together with updated value. On receiving this invalidation confirmation the CMs updates the entry UC to indicate the confirmed update in the DS. The updated cached copy of the data item is immediately available with a change in PLP. This ensures cache consistency with control of concurrency.

The PLP of the data item is calculated as UI (D_{id})-($U(D_{id})$ * $UI(D_{id})$) where UI is the actual update interval of the data item and U is the update rate on the data item. Update rate is the ratio of the number of transactions that have updated the data item to the total number of affiliated transactions and is calculated as:

$$U(D_{id}) = Total\_Updates\ (D_{id})/ATC\ (D_{id})$$

Since the PLP is based on the update rate, it is highly probable that the PLP is very close to the actual valid life span of the data item. If the PLP is optimally estimated, the number of transactions that need to be restarted will be considerably reduced. Since the predicted life time is based on update rate, the value of PLP is dynamically changed reflecting the data update environment.

The proposed consistency preservation technique also deals with the MH that has missed IR. When a MH involves in data processing even after IR miss, the cache manager checks whether the data already read is valid or not by comparing the timestamp of the last read operation and last update timestamp. If invalid data item, the CM sends the updated value to the MH.

Proposed ATC/DTL based replacement policy: In the proposed approach, ATC indicates the degree of sharing of the data item. Analysing the previous approaches, none of them considers the number of transactions that share the data item. If more number of transactions share the same data item, it is highly desirable to retain the data item in the cache. Because,

if the highly shared data item is evicted from the cache, it is highly probable that the cache hit ratio of the affiliated transactions becomes zero. Since the proposed framework favours the concurrent transactions, proposed policy replaces the data object with the least ATC. When more than one data item has the same ATC value, it is resolved based on recency of the usage and the count of dependent transactions on the data item. In the proposed ATC/DTL based policy, highly shared recent data items are favoured and retained in the cache.

RESULTS

Impact of update rate on PLP: The PLP of the data item for various update intervals and update rates are tabulated in Table 1. When 80% of the affiliated transactions involved in data update, the PLP is reduced to 20% of the previous update interval.

Impact of PLP on usage of invalid data: The accuracy of the predicted life time of AVI/TTL and PLP is shown in Table 2.

Table 1: Impact of Update Rate on PLP

Update Interval in s	Predicted life period in s U = 0.2	U = 0.4	U = 0.6	U = 0.8
5	4.0	3.0	2.0	1.0
4	3.2	2.4	1.6	0.8
3	2.4	1.8	1.2	0.6
2	1.6	1.2	0.8	0.4
1	0.8	0.6	0.4	0.2

Table 2: Accuracy of PLP and AVI/TTL Calculation

Actual UI in sec	Maximum Deviation in PLP/AVI/TTL calculation (sec) U=0.8, PLP = 1, AVI=5 and TTL=3.5 for UI=5 PLP	AVI	TTL	U=0.6, PLP = 2 AVI=5 and TTL=3.5 for UI=5 PLP	AVI	TTL
5	0	0	0.0	0	0	0.0
4	0	1	0.0	0	1	0.0
3	0	2	0.5	0	2	0.5
2	0	3	1.5	0	3	1.5
1	0	4	2.5	1	4	2.5

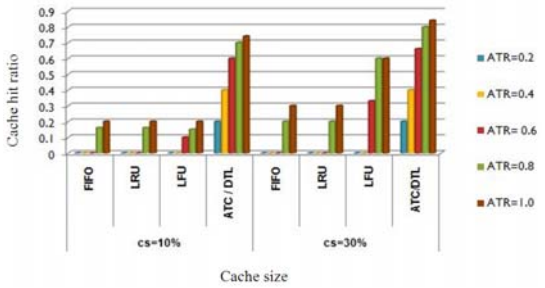

Fig. 2: Analysis of cache hit ratio

Analysis of cache hit ratio: The cache hit of the proposed ATC/DTL method is compared against existing FIFO, LRU, LFU methods and shown in Fig 2. The cache hit ratio of the proposed policy is improved by at least 20% when compared to the existing cache replacement techniques.

DISCUSSION

Impact of update rate on PLP: From the Table 1, it is understood that, since the increase in update rate reduces the PLP of the data item, it is highly probable that the PLP equals the actual valid life time of the data.

Impact of PLP on usage of invalid data: In AVI (Moiz and Nizamuddin 2008) approach, the predicted validity period of the data item is equal to the previous update interval. The SACCS (Wang *et al*., 2004) method predicts TTL value based on previous update interval and life time. AVI/TTL values are predicted based only on the update interval irrespective of the update rate. But the proposed PLP based scheme calculates the PLP, based on the update rate. From the Table 2, it is clear that, in AVI/TTL approach the validity lifetime deviates whenever the update rate is changed. The PLP approach ensures no deviation when update rate is uniform and update rate is not increased more than half of its previous update rate.

Analysis of stale hit probability: Stale Hit Probability is the probability with which the MH accesses the invalid (stale) data from the cache. Since cache is maintained in the MH, the AVI and SACCS approach assume validity of the data till AVI/TTL expires. In SACCS approach, the stale hit probability is proportional to the IR miss probability. The stale hit probability reaches 0.51 when IR miss probability increases to 0.8 irrespective of the update rate. But in the proposed PLP method the stale hit probability is reduced to zero as the updated data is made immediately available in the cache.

Impact of validity period on transaction 'restarts': In SACCS approach, when MH is awake the data item is assumed to be valid before TTL expiration. The probability of transaction 'restarts' due to stale data access, is directly proportional to stale hit probability. For AVI based approach, the transaction is committed locally and conflict resolution is postponed till conflicts are identified in the coordinator. The coordinator restarts all the transactions that processed with stale data. But in the proposed approach no invalid data is processed and the conflicts are resolved even before the transactions reach the commit point. The proposed PLP based technique reduces the transaction 'restarts' to 0

when PLP exactly matches the actual life time. The existing techniques abort all the transactions except the one that commits first.

Comparison of cache hit ratio: Since the proposed ATC/DTL based cache replacement policy assumes data items of equal size, it is compared only against traditional approach in which data size is equal. From Fig. 2, it is understood that the cache hit ratio of the proposed method increases as ATC ratio increases irrespective of the cache size. The cache hit ratio is improved by 20% as highly shared data items are retained in the cache. It ensures increased cache hit ratio not only for increased ATC ratio but also reduced ATC ratio. The results prove that the proposed method favors concurrent transaction execution than existing techniques.

CONCLUSION

The proposed PLP approach calculates PLP based on the update rate. It ensures no deviation when update rate is uniform and update rate is not increased more than half of its previous update rate. The PLP method reduces the stale hit probability and number of transaction 'restarts' to 0 when PLP exactly matches the actual life time. The cache hit ratio of the proposed ATC/DTL based replacement policy is improved by at least 20% when compared to the existing techniques. It also proves that the replacement policy is highly suitable for concurrent transaction execution environment than the existing policies. The proposed strategy supports concurrent public data processing in mobile wireless environment.

REFERENCES

Barbara, D. and T. Imielinski, 1994. Sleepers and workaholics: Caching strategies in mobile environments. Proceedings of the SIGMOD International Conference on Management of Data, (SIGMOD'94), ACM New York, NY, USA., pp: 1-12. DOI: 10.1145/191839.191844

Kahol, A., S. Khurana, S.K.S. Gupta and P.K. Srimani, 2001. A strategy to manage cache consistency in a disconnected distributed environment. IEEE Trans. Parallel Distributed Syst., 12: 686-700. DOI: 10.1109/71.940744

Wang, Z., S.K. Das, H. Che and M. Kumar, 2004. A Scalable Asynchronous Cache Consistency Scheme (SACCS) for mobile environments. IEEE Trans. Parallel Distributed Syst., 15: 983-995. DOI: 10.1109/TPDS.2004.60

Moiz, S.A. and M.K. Nizamuddin, 2008. Concurrency control without locking in mobile environments. Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology, July 16-18, Nagpur, Maharashtra, pp: 1336-1339. DOI: 10.1109/ICETET.2008.182

Yin, L., G. Cao and Y. Cai, 2005. A generalized target-driven cache replacement policy for mobile environments. J. Parallel Distributed Comp., 65: 583-594. DOI: 10.1016/j.jpdc.2004.12.002

Chand, N., R.C. Joshi and M. Misra, 2006. Efficient cache replacement in mobile environment using data profit. Proceedings of the 12th International Conference on Parallel and Distributed systems, (ICPADS'06), IEEE Computer Society Washington, DC, USA., pp: 203-212. DOI: 10.1109/ICPADS.2006.39

Hiary, H., Q. Mishael and S. Al-Sharaeh, 2009. Investigating cache technique for location of dependent information services in mobile environments. Eur. J. Sci. Res., 38: 172-179.

Shi, S., J. Li and C. Wang, 2005. A new semantic cache management method in mobile databases. J. Comput. Sci., 1: 351-354. DOI: 10.3844/jcssp.2005.351.354

Elfaki, M.A., H. Ibrahim, A. Mamat and M. Othman, 2011. Collaborative caching architecture for continuous query in mobile database. Am. J. Econ. Bus. Admin., 3: 33-39. DOI: 10.3844/ajebasp.2011.33.39