Multiple Servers - Queue Model for Agent Based Technology in Cache Consistence Maintenance of Mobile Environment

G.Shanmugarathinam[1], Dr.K.Vivekanandan[2]

[1] Research Scholar(external mode) in Bharathiar university Coimbatore, Tamil Nadu,India

[2] Professor, Bharathiar university Coimbatore, Tamil Nadu,India

Abstract

Caching is one of the important techniques in mobile computing. In caching, frequently accessed data is stored in mobile clients to avoid network traffic and improve the performance in mobile computing. In a mobile computing environment, the number of mobile users increases and requests the server for any updation, but most of the time the server is busy and the client has to wait for a long time. The cache consistency maintenance is difficult for both client and the server. This paper is proposes a technique using a queuing system consisting of one or more servers that provide services of some sort to arrive mobile hosts using agent based technology. This services mechanism of a queuing system is specified by the number of servers each server having its own queue, Agent based technology will maintain the cache consistency between the client and the server .This model saves wireless bandwidth, reduces network traffic and reduces the workload on the server. The simulation result was analyzed with previous technique and the proposed model shows significantly better performance than the earlier approach.

Keywords: *mobile database, wireless networks, database cache, Queuing model.*

1. Introduction

Mobile Computing uses portable computing devices such as laptops, PDAs and wearable computers. Example applications used in mobile computing environment include sales force automation, order entry, e-mail, calendar management, financial and news services, insurance companies, emergency services (police, medicals), traffic control, taxi dispatch, etc.

A mobile computing environment is a distributed system, thus when data at the server changes, the client hosts must be made aware of this fact in order to invalidate their cache, otherwise the host would continue to answer queries with the cached values returning incorrect data.

Mobile computing has stringent constraints in network resources, such as bandwidth and connectivity. As such, data in mobile applications are often cached at clients to increase performance, data availability and reliability. Most fault-tolerant schemes for wireless sensor networks focus on power failures or crash faults. Little attention has been paid to the data inconsistency failures

Recent advances in wireless and mobile networks have led to the exponential growth of mobile applications although a number of studies have been made in this subject, few researchers focused on mobile data access. In this design a node as middle server (MS). It is between the server and client. Whenever server data was updated immediately synchronization starts with Middle server and the client. Some of the clients wake up from sleep mode immediately request the Middle server for the updated data and need not request the server. So it reduces the work load in the main server database.

2. Related work

2.1 Updated Invalidation Report (UIR) based cache techniques.

In this approach [1] Invalidation Report based cache management is an attractive approach for mobile environments. In this approach the server periodically broadcasts an IR in which the changed data item are indicated. Since IR arrive periodically, client can go to sleep most of time and only wake up when the IR comes. It has some drawbacks such as long query latency and low hit ratio. There is a long latency problem with a UIR (Updated Invalidation Report) based approach, where a small fraction of the essential information related to cache invalidation is replicated several times within an IR interval, and hence the client can answer a query without

waiting until the next IR. If there is a cache miss the client still needs to wait for the data to be delivered.
It has some drawbacks such as long query latency and low hit ratio. There is a long latency problem with a UIR (Updated Invalidation Report) based approach.

2.2 Adaptive Energy Efficient Cache Invalidation Scheme

In this approach [2] to reduce the bandwidth requirement, the server transmits in one of three modes slow, fast and super-fast .The mode is selected based on thresholds specified for time and the number of clients requesting updated objects.
The mode is selected based on thresholds specified for time and the number of clients requesting updated objects. The updating is less in server if the mode changes to slow, so the client has to wait for long time to utilize cache data during invalidation report

2.3 Smart Server Update Mechanism (SSUM)

In this approach [3] SSUM the server send data updates to the cache node (CN). Request Mobile host that desires a data items sends its request to its nearest query director (QD). If this QD finds the query in its cache, it forwards the request to the CN caching the items, which in turn sends the item to the requesting mobile host. Otherwise it forward it to its nearest QD, If the request traverses all QDs without being found, a miss occurs and it gets forwarded to the server which sends the data item to the request mobile host
In this case, latency is more as well as bandwidth is wasted because of this the client should wait idle for the server to reply.

3. Queuing model for Agent based technology

In this paper proposed Queuing model for Agent based technique for cache consistency for wireless network .In our design does not required to produce an Invalidation report, In mobile computing the mobile user increase and request the server for any updating but most of the time server is busy and client to wait for long time .so we design the node as middle server using between server and client, thread agent maintain a log and thread synchronization in client and server, maintain the cache consistency .
The following subsection describe the proposed algorithm in detail

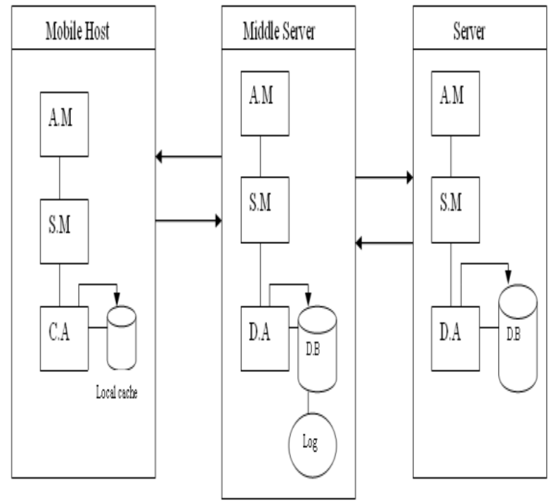


Fig 1: System Architecture

AM –Agent manager

S.M- Security module

C.A-Cache Agent

D.A-Database agent

DB-Database

3.1 Middle server with Agent based technique

Some of client data cache is placed in Middle server. To find the middle server it is based on the network bandwidth, speed of CPU, lower latency , cache hit
Ratio The middle server are near to the client, so the communication cost, energy consumption are very less, easy to update the cache data and easily maintain Consistency. If Data requested is not available in local cache, the client send the broadcast request to the middle server . Middle server receive the packet , search in the cache and send the acknowledge data to the client. The Middle servers satisfy the nearest client request. Advantage is low cost for communication and reduces the network traffic in mobile networks. The Middle server log and Agent synchronization Model maintains consistency between the Server data and Middle server. For each cached data object uses log to maintain consistency between Server, Middle server and Mobile client. When a data $d_{x\ is}$ retrieved by a mobile client log is created to indicate data is valid or not.
If and when the Server receives an updated data object d_x it broadcasts and synchronizes with Agent Manager of client to make cache data object reliable. During this process a log maintained in server is compared with recent log of Middle server If so there is a need of Updation, it processes to perform update function(s). In mobile environments a Mobile Cache is one of two states. (i) Awake or (ii) Sleep. If a Mobile Client is awake an

IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013
ISSN (Print): 1694-0784 | ISSN (Online): 1694-0814
www.IJCSI.org

internal request is shared between Agent Manager at Middle server and Agent Manager at client to ensure that data object is updated. If there is an Updation the SynchM of server synchronizes with SynchM of Middle server and client in order to make as valid data object.
The data objects of a Mobile Client in the sleep state are unaffected until it wakes up. When a mobile client wakes up a new Agent upon is created which holds last accessed log, this log passes to the Middle server, On receiving upon the log it compares with previous log maintained by it. If it is invalid data cache the Agent Manager of middle sever cache starts the synchronizes with client for the updating data.

3.2 Queuing system

A queuing system consist of one or more server that provide services of some sort to arriving .Every day examples can be described as queuing systems ,such Computer system , manufacturing systems ,maintenance systems, communication system. This services mechanism of a queuing system is specified by the number of server each server having its own queue or common queue. The essence of queuing theory is that it takes in to account the randomness of the arrival process and the randomness of the service process. The word randomness refers to the probability distribution used in the arrival as well as service process.
One of the most important queuing models is the Evlang loss models, it assumes that arrivals follows a passion process and that the blocked Mobile host (those who find all server busy) are cleared (that is they are denied entry in to the system so the blocked Mobile Host are lost)

The probability of blocking is Evlang B formula given by

$$B(s,a) = \frac{\frac{a^s}{s!}}{\sum_{k=0}^{s} \frac{a^k}{k!}} \qquad (1)$$

$$a = \lambda t$$

Where "S" is the numbers of server
"a" is equal to λτ where
"λ" is arrival rate and
τ is the average services time

B(1,0.8)=0.444
S0 44.44% of the arrivals will be blocked
Similarly increase the number of server s the percentage of blocking can be computed from Evlang B formula . but when s and a are large it is hard to calculate directly so interactive scheme or formula is designed as follow :

$$B(n,a) = \frac{a\ B\ (n-1,a)}{n + a\ B\ (n-1,a)} \qquad (2)$$

$$B(10,8) = \frac{B(9,8)}{10 + 8\,B(9,8)}$$

$$B(10,8) = \frac{8^{10}}{10!} \div \sum_{k=0}^{10} \frac{8^k}{k!}$$

$$B(10,8) = \frac{8^{10}}{10!} \div e^8$$

$$B(10,8) = 0.1217$$

$$B(10,8) = 0.1217$$

This means that when 8 Evlangs of passion traffic is offered to 10 server then about 12 % of the arrivals will be blocked . since the blocked mobile host are cleared from the system .

Table 1 : Shows the Blocking percentage of Mobile Host

No of Server	Blocking Percentage
1	44.4
2	15.09
3	3.86
4	0.76
5	0.12
6	0.016
7	0.0018
8	1.8696652894159525E-4
9	1.6619244255037438E-5
10	1.3295395227262418E-6

4. Performance Evaluation and Results Discussion

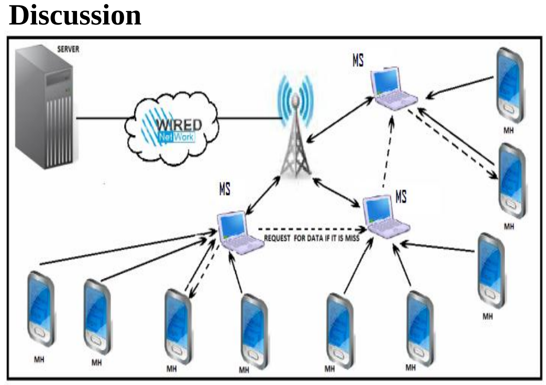


Fig 2: Simulation for Agent Based Model

In simulation created a Mobile Sales, online distributor of clients for interact with a sales system. The workload consists of a set of interactions: number of clients can be added , create new orders, change payment types, check on the status of previous orders, view new products the distributor might have recently added, look at detailed descriptions of products, and make changes to the product log. Each client's test run is defined by a time period that specifies the length of execution. The proposed algorithm compare with SSUM algorithm and the result shows the reduce the average access latency (seconds) .

Client cache Client configuration Client View Detail

Fig 3: simulation of mobile Host

Client sleep mode client cache updating client wake up

Fig 4: simulation of mobile Host

ID	Symbol	ISIN	CPrice
1	BHEL	INE257A01026	789.23
2	AXISBANK	INE238A01026	1070.70
3	HCLTECH	INE860A01027	570.55
4	TATASTEEL	INE081A01012	404.54
5	TCS	INE467B01029	950.56

Fig 5 : Middle server Database

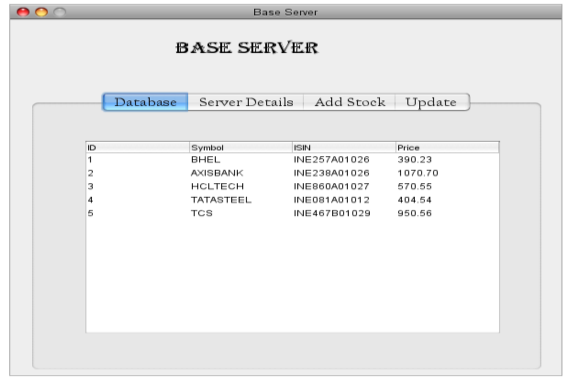


Fig 6: Base Server Database

Client_Id	Symbol	Price	Updated
3	BHEL	420.34	20/09/2012 01:20:00 PM
1	BHEL	420.34	20/09/2012 01:20:00 PM

Fig 7: Middle server Log

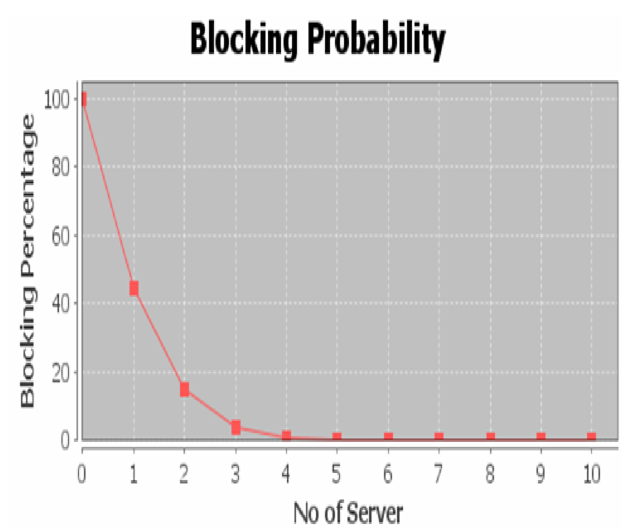
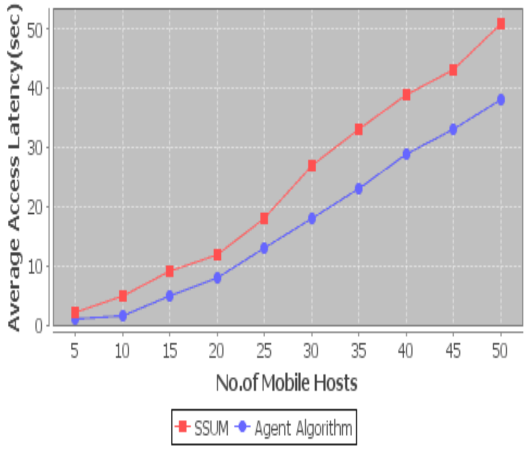


Fig 8: Number of server vs Blocking percentage

Above the graph it shows increase the number of servers, the blocking percentage of server services was reduced



.

Fig 9 : Comparing Agent Algorithm with SSUM Algorithm

5. Conclusion

In this paper, the proposed technique of multiple server queuing system reduces the blocking percentage of server services. key features as stated earlier, are low cost for communication, using a Agent based Algorithm which is easy to maintain the cache consistency and reduction of network traffic . Simulation results shows that the Agent based algorithm gives a significantly better performance than earlier approaches.

Reference

[1] “Niraj tolia and adam wolbach, improving mobile a database Access over wide area networks without degrading consistency ACM 2006

[2] “Alok Madhukar, Reda Alhajj, “An Adaptive Energy Efficient Cache Invalidation Scheme for Mobile Databases”, ACM 2006, April 23-27, 2006

[3] “Khaleel Mershad ,Hassan Artail, “Smart Server Update Mechanism for Maintaining Cache consistency in Mobile Environments” IEEE Transaction on Mobile Computing Vol 9, No 6 June 2010

[4] Huaping Shen, Mohan Kumar, Sajal K.Das, ”Energy Efficient Caching and Prefetching with Data Consistency in Mobile Distributed Systems”, 0-7695-2132-0/04/2004, IEEE.

[5] Yu Huang, Jiannong cao, Beihong Jin, Xianping Tao. “Flexible Cache consistency Maintence over Wireless Adhoc Networks” IEEE Transaction on Parallel and Distributed System Vol 21,No.8,August 2010

[6] 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.

[7] Elsen, F. Hartung, U. Horn, M. Kampmann, and L. Peters, “Streaming Technology in 3G Mobile Communication Systems”, Computer, 34, No. 9, pp. 46-52, 2001.

[8] S. Hosseini-Khayat, “On Optimal Replacement of Nonuniform Cache Objects”, IEEE Trans. Computers, 47, No. 4, pp. 445-457, 2000.

[9] T. Imie linski and B.R. Badrinath, “Wireless Graffiti— Data, Data Everywhere Matters”, Proc. 28th Int’l Conf. Very Large Data Bases, 2002.

[10] 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, 12, No. 7, pp. 686-700, 2001.

[11] Mark Kai Ho Yeung, Yu-Kwong Kwok, “Wireless Cache Invalidation Schemes with Link Adaptation and Downlink Traffic”, IEEE Transaction On Mobile Computing, 4, No 1, January/February 2009

First Author : G. Shanmugarathinam has been teaching in Engineering College for more than 12 years. He is a Research Scholar(external mode) in Bharathiar university in Coimbatore, Tamil Nadu,India He has been the author of various books and papers which he has published more than 5 articles in international journals. His fields of interest are Mobile Computing, Cisco networking, Management of information system, wireless network and network security.

Second Author : K.Vivekanandan is a Professor , school of management at Bharathiar University, He has 5 years industry experience and 26 years of teaching Experience .His research interests include Management Information System, E-marketing, Quantitative Methods , Data Mining and mobile computing. He has published more than 30 articles in national/international journals and he has successfully guided 16 PhD’s and is currently guiding 7 students.