[NC-Rase 18]
DOI: 10.5281/zenodo.1884722

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

EFFECT OF CACHE PERFORMANCE ON EFFICIENT DATA RETRIEVAL IN MOBILE NETWORKS

K.R. Patil*1, Dr.V. M.Patil2 & Dr. V.M. Thakare3
*1D.C.P.E., H.V.P.M., Amravati(Mah.), India
2H.O.D. (Comp. Dept), S. S. S.College, Akola(Mah),India
3H.O.D. (Comp. Dept), S.G.B.A.U., Amravati(Mah.),India

ABSTRACT

Due to advancement in wireless network and its facilities, people became able to use mobile computers (Laptops, palmtops, etc.) to do various computing tasks and to access the Internet from anywhere and anytime. The development of mobile computing has made this goal possible. But still there are lots of problems regarding connections and faster access to required information. Since mobile devices are resource limited in terms of memory, storage, display area, power, etc. and also a wireless link has low bandwidth and unstable. All these factors cause problems for mobile applications which results in bad performance. One of the efficient approaches to reduce communication overhead is to design a continuous answer maintenance scheme that allows the user to collaborate with peers to continuously maintain the answer which bypasses the always processing of the query from scratch, in order to maintain QoS[1]. The technique used for answer maintenance is catching with improved performance.

Keywords: *caching, semantic caching, hit ratio.*

I. INTRODUCTION

Mobile environment encompasses the mobility of hardware, communication among the moving devices for access to data as well as software residing on these mobile devices. Various kinds of queries are to be used to access data from mobile hosts. These queries may be location queries, continuous queries, existence queries and data queries [2]. In mobile ad hoc networks, nodes communicate with each other by means of broadcast radio signals where all nodes in the network get the broadcasted message [3]. The wireless channel is shared among adjacent hosts and network topology can always change as hosts move. The mobile query processing splits their computations into sets of operations of which some operations get execute on a mobile host while others get executed on stationary host. Disconnection is the main problem of mobile computing. To overcome this transaction shares their states and partial results with other transactions [4]. As the mobile hosts move from one cell to another, the location information, the states of query processing, states of accessed data objects also move. For better performance, the query processing architecture should tackle the limitations of mobile computing like limited battery life, low bandwidth, disconnection and reduced storage capacity. Caching of frequently accessed data at the mobile clients has been shown to be a very useful and effective mechanism in handling this problem. Caching improves the efficiency and reliability of data delivery over the network.

II. APPLICATION DOMAIN

Applications for mobile computers can be divided into the following five categories:

1. Standalone applications such as games or utilities
2. Personal productivity software (word processors, presentation software, calendars);
3. Internet applications such as email, WWW browsers
4. New location-aware applications such as tour planners and interactive guides.
5. Ad-hoc network and groupware applications

The applications getting the most attention are the ones that best take advantage of the mobility of wireless Internet devices, and which suffer least from the limitations of wireless platforms

There is still a lot of time we are mobile – moving among offices, homes, planes, trains, automobiles, conference rooms, and classrooms and such. There is need to access computing resources not only when stationary but also while mobile or in transit.

As technology improves For example, people can send and receive emails and access Internet web sites using mobile computers via wireless networks. The future trend of telecommunication is to extend the telecom and computing services to mobile users, to break the restriction of user locations, to allow people access to computing resources anywhere and anytime at will. Mobile devices are resource -poor relative to static elements. Wireless links have low bandwidth and are unstable. Mobile elements must operate under a much broader range of networking conditions. The nature of wireless communication media and the mobility of computers combine to create fundamental new problems in networking, operating systems, and information system.

III. HOW TO IMPROVE PERFORMANCE OF MOBILE APPLICATIONS

Mobile distributed systems raise new issues such as mobility, low bandwidth of wireless channels, disconnections, limited battery power and lack of reliable stable storage on mobile nodes. This results in mobile computing applications to suffer from bad performance. Hence mobile users suffer from long latency. The effort of improving performance of mobile computing systems contains every aspect of mobile environment, from hardware to software, from proper network configuration to efficient application design. In order to cope up the resource constraints in mobile computing many techniques for optimizing data transfer over wireless links are proposed and utilizes mobile device resources fully. The commonly used techniques include caching, prefetching and compression. Each technique has its strength and weakness. The major criteria to check the performance are user perceived latency, request meet rate, cache hit rate, and wireless bandwidth efficiency.

IV. WHAT IS CACHING

Since the bandwidth of the wireless link is very limited, it becomes crucial to minimize wireless communication in order to reduce the contention for the narrow bandwidth. To handle this problem cache the data that is frequently accessed at the mobile clients. It proves to be very useful and effective mechanism as caching improves the efficiency and reliability of data delivery over the network. When a mobile client makes a request for certain data, it first checks the cache, if the required information is there, the client replies with it; otherwise, the client fetches it from the information server. This data is used to answer the given query and again get stored as a copy in the cache as well for future use. In this way a cache in the mobile client is useful to serve a user’s request quickly and reduce user perceived latency. This mechanism reflects the usefulness in energy savings and cost savings.

A caching mechanism consists of the following aspects: cache size and replacement strategy. Since cache storage is limited in mobile devices, what to cache, when to cache have to be carefully considered. It is trivial to decide how long to cache the information and how to replace the cached datafor future use. It is seen that file size and data types serves an important role in the caching policy. A cache system with a higher hit ratio and a lower query processing time is considered to be an efficient cache system. Semantic cache has a potential to maximize the hit ratio as compared to other cache organization techniques because it has an ability to answerfully as well as partially overlapped queries.

V. ISSUES RELATED TO CACHING

1. Need for caching

If the file is not dynamic and there is no space, in such situation the file should always be cached. When space is insufficient, there are three simple strategies for deciding which files to cache: "cache all", which removes other files to make space; "threshold", the same as the previous strategy, but only caching files below a certain size; and "adaptive dynamic threshold," whereby the maximum file size threshold alters dynamically[5]. Cache is called hit if data is found in it and hit ratio is the percentage of user posed queries that can be answered locally (partially or fully) from the cache. Therefore, the cache system should be designed in such a way that it increases the hit ratio.

ISSN 2348 – 8034
Impact Factor- 5.070

Improvement in hit ratio ensures reuse of stored data, lesser amount of data is required to be retrieved from remote locations[6].

2. Consistency maintenance

While applying caching to the application, Cache consistency maintenance is an important issue to consider. Due to limitation of battery power, mobile computers may often forced to operate in a sleep or even totally disconnected mode. As a result, some cache invalidation reports broadcasted by a server may be missed by the mobile computer. It results in making the cached file out -dated or forces it to discard the entire cache contents after waking up.

3. Fault tolerance using cooperative caching

To cope with communication bandwidth and storage constraints, cooperative caching [7] method prioritizes the data-items in terms of their value, as reflected by supply and demand. Performing Simulations based on real-life mobility traces. This approach proves to be better than existing cooperative caching strategies and an existing mobile sensor network algorithm.

The model using cooperative caching uses MARKET algorithm [7] for querying mobile P2P databases. A novel strategy used in MARKET for a mobile peer to prioritize the reports based on their relevance. The relevance of a report depends on its size, demand (how many peers are querying it), and supply (how many peers already have it). Queries are disseminated to enable the estimation of demand. A machine learning algorithm, called MALENA, is used to enable the estimation of the supply.

VI. USE OF CACHE IN QUERY PROCESSING

Query trimming (*Qtr*) and query rebuilding (*Qrb*) are two major activities in query processing. The process of query trimming splits the user query into probe and remainder queries[6]. The query rebuilding process merges the results of probe and remainder queries. Query trimming process is further divided into two basic steps. First, semantics of the newly posed query are matched with the stored semantics on cache, which is called query matching (*Qm*). The query matching process checks whether the data is present in cache. An efficient query matching process finds all the hidden semantics out of the stored semantics in cache. These finding results in increasing hit ratio. Optimum query matching ensures optimum query trimming.

In caching a MH maintains a time-stamp[8] for its cache which is the time-stamp of the last invalidation report/data that it had successfully received from the server. Due to several disconnections we do not know the time at which the MH got disconnected and just by using time stamp we can handle both failures and voluntary disconnections. Even if the MH wakes up and then immediately goes to sleep before receiving the invalidation report, maintaining consistency of the cache becomes a crucial task as it would use the old value of time stamp . To maintain correct data in cache, proper cache maintenance and replacement strategies are important so it get the correct information in the invalidation report. Thus, arbitrary sleep patterns of the MH can be easily handled[9].

Whenever the MH regains network connectivity and makes the first query after reconnection it informs the MSS of its cache time-stamp. The MSS refers to the MH's HLC and compares the advertised cache time-stamp with the time-stamp of each data item in the HLC. All data items that have changed while the MH was disconnected are collected and invalidation report is sent to the MH. This way an MH can determine what data items had changed while it was disconnected. In order to maintain consistency, the MH has to defer all queries until this exchange has been completed. An energy efficient cache invalidation method GCORE allows a mobile computer to operate in a disconnected mode to save battery while still retaining most of the caching benefits after a reconnection. GCORE[10] can substantially improve mobile caching by reducing the communication bandwidth and energy consumption for query processing.

VII. CACHE REPLACEMENT SCHEMES

Caching can reduce the bandwidth requirement in a mobile computing environment. It is also useful in battery failure situation where a wireless mobile computer may often be forced to operate in atotally disconnected mode.
Cache replacement process is tightly related to the cache size: if an infinite cache exists, there is no need for such a file removal process since all the files that enter the cache are kept forever. In reality, the cached files have to be replaced because of a lack of memory space, and the choice of the next file to be removed is quite important. When deciding which file to discard, the key point is to discard the file that is most unlikely to be used again. Factors that impact cache performance include: number of references to the files, the file size, and the file time -to-live, and file retrieval time.

File replacement strategies have a great influence on cache performance. The replacement process is tightly related to the cache size: if an infinite cache exists, there is no need for such a file removal process since all the files that enter the cache are kept forever. In reality, the cached files have to be replaced because of a lack of memory space, and the choice of the next file to be removed is quite important. While deciding which file to discard, the key point is to discard the file that is most unlikely to be used again.

Factors that impact cache performance include: number of references to the files, the file size, and the file time -to-live, and file retrieval time. For example, consider the factor of file size. For all caches, an upper limit is defined for the size of the caching area. The file replacement strategy can be split in two phases: first, the documents are sorted in order to determine the removal order; second, one or more documents are removed from the removal list. Cache replacement strategies can be divided into two groups: on -demand replacement strategies and periodic replacement strategies. In an on -demand replacement strategy, the removal process is started once the cache is full or the remaining amount of memory is not enough to store the new incoming file. In a periodic replacement strategy, files are removed after a certain amount of time. The possible disadvantage of the on -demand replacement strategy is that if a cache is nearly full, then running the removal policy only on -demand will invoke the removal policy on nearly all document requests, if removal is time consuming, it might generate a significant overhead. On the other hand, periodic removal reduces the hit -ratio because files are removed earlier than required. In the following we study different cache replacement strategies.
1. Random replacement: A random replacement simply discards a randomly chosen file.
2. The least recently used (LRU): The simplest and most common used cache management approach, which removes the least recently accessed files until there is sufficient space for the new file. This approach not only stores data also the time that each file was last referenced. LRU is widely used as a replacement policy. It is well understood and shown to perform well in a wide variety of workloads.
3. The least frequently used (LFU): LFU keeps track of the number of times a file is used and replaces the least frequently used file to make room for an LFU is more complex to implement than LRU. In LRU, a referenced file is just placed at the head of removal list. In LFU, the referenced file needs to be placed in a sorted (by frequency of use) list and this in general is more complex. One modified version of LFU is LFU-aging. LFU-aging allows an incoming file to replace only blocks with a frequency count of 1. First-in-first-out (FIFO): discards the file that is the oldest in the cache. It is easily implemented as a circular buffer.
2. Largest-file-first (LFF): discards the largest file in the cache.

Cooperative Cache Replacement
One of the important cache replacement techniques is update based cache replacement strategy [11]. This can be further divided into least access to update (LA2U) and least access to update difference (LAUD). LA2U and LAUD are based on update frequency and access frequency respectively. These techniques concludes that update based cache replacement technique improve the cache performance significantly. This technique is preferred at the situation when the network experience heavy updates. To decrease the cache miss ratio, a special cache replacement strategy is designed called time and distance sensitive (TDS) [12]. As the name indicates, it is a function of both time and distance. Distance is a measure of number of hops. Time is also considered as the distance becomes obsolete.

ISSN 2348 – 8034
Impact Factor- 5.070

VIII. CONCLUSION

Since the bandwidth of the wireless link is very limited, it becomes crucial to minimize wireless communication in order to reduce the contention for the narrow bandwidth. To handle this problem cache the data that is frequently accessed at the mobile clients. It proves to be very useful and effective mechanism as caching improves the efficiency and reliability of data delivery over the network. Caching can reduce the bandwidth requirement in a mobile computing environment. Due to battery power limitations, a wireless mobile computer may often be forced to operate in a totally disconnected mode. **As** a result, the mobile computer may miss some cache invalidation reports broadcasted by a server which forces it to discard the entire cache contents after waking up. To overcome this problem cache replacement scheme and prefetching schemes are integrated together to get the maximum usage of cache. Different- criteria policies are more successful than single- criteria policy at deciding which objects to cache.

REFERENCES

1. *Ing Ray Chen, AnhPhanSpeer, Mohamed Eltoweissy,"Adaptive Fault-Tolerant Qos Control Algorithms For Maximizing System Lifetime Of Query-Based Wireless Sensor Networks", IEEE transactions on dependable and secure computing, Vol. 8, No. 2, pp.161-176, March-April 2011.*
2. *Aymen Al-Ani , Jochen Seitz, "An approach for QoS-aware routing in mobile ad hoc networks",IEEEInternational Symposium on Wireless Communication Systems (ISWCS), 2015 ,25-28 Aug. 2015*
3. *M.VijayaLakshmi, Dr.D.SreenivasaRao, " A comparative analysis on Multicast routing protocols in Mobile Ad-hoc networks : MAODV, ODMRP, PUMA & AM Route", Global journal for research analysis, Vol.4, Issue 8, pp 347 -349, August 2015.*
4. *Ahmad Al-Qerem, "Performance evaluation of transaction processing in mobile data base systems", International Journal of Database Management Systems (IJDMS), Vol. 6, No. 2, April 2014.*
5. *Sanjay Kumar, SandhyaUmrao, "Improve Client performance in Client Server Mobile Computing System using Cache Replacement Technique", International Research Journal of Engineering and Technology (IRJET), vol. 05 Issue 03, pp 3164-3177, Mar2018.*
6. *Munir Ahmad, M. Abdul Qadir and Tariq Ali, "Indexing for semantic cache to reduce query matching complexity", Journal of the National Science Foundation of Sri Lanka 45(1),pp 13-22.*
7. *Bo Xu, FatemehVafaee, OuriWolfson, "In-Network Query Processing in Mobile P2P Databases", ACM GIS 09, pp 4-6, November 2009.*
8. *Ahmed Elmagarmid, Jin Jing, Abdelsalam (Sumi) Helal, and Choonhwa Lee, "Scalable Cache Invalidation Algorithms For Mobile Data Access", IEEE transactions on knowledge and data engineering, Vol. 15, No. 6, pp. 1498-1511, November/December 2003.*
9. *Kun-Lung Wu, Philip S. Yu and Ming-Syan Chen, "Energy-Efficient Caching for Wireless Mobile Computing", 1063-6382/96 $5.00 ,1996IEEE, pp 336-343.*
10. *ChandraniChakravorty and Dr. Usha J., " Cache Management Issues In Mobile Computing Environment", International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol.2, No.1,PP 21-27 February 2012*
11. *B. Zheng, J. Xu, and D. Lee."Cache invalidation and replacement strategies for location-dependent data in mobile environments". IEEE Transactions on Computers, 51(10):1141. 1153, October 2002.*
12. *AnuragKahol, Sandeep K. S et . al , "A Stretegy to manage cache consistency in a disconnected Distributed Environment " ,IEEE Transactions on Parallel and Distributed System, Volume 12 Issue 7,July 2001.*