

A COMPARATIVE STUDY OF SEMANTIC CACHE TECHNIQUES IN MOBILE DATABASE ENVIRONMENT

S. Qadri, M.P. Tariq and M. Shoaib

Department of CS VU, Lahore, Pakistan, Department of CS & E, UET Lahore, Pakistan

Corresponding author: salmanbzu@gmail.com

ABSTRACT: For the last few years, database accessibility is a great issue especially in the distributed database environment. Recently mobile computing is becoming a hot issue. Researchers are trying to find out the basic factors which affect the performance of semantic cache implementation strategies. It is also tried to find out the methodology, how the issues of accessibility and reliability of mobile database can be solved in the best and efficient way. The authors of this research work present a comparative study of semantic intra file, semantic inter file and fully associative techniques in detail. These techniques help in resolving the issues of accessibility and efficiency of mobile database. A Simulator is used to generate results of these three techniques. Statistical package for social sciences, a software tool is used to show statistical data analysis and comparison of these techniques in the form of graphs. This study shows that semantic intra file is the best technique.

Key words: Semantic cache, Mobile Database, LRU, Simulator

INTRODUCTION

Mobile database has many problems like unreliability of connection, limited storage capacity, and security and privacy issues. Unreliable connection creates problems while accessing data from the server. Security and privacy issues increase the chances of theft and unauthorized access to data. Limited storage capacity makes it difficult to store large data on the mobile computers (Darin and John, 2003). Mobile computing, a new paradigm helps users to access data and information services without caring their actual location and movement behavior. This is all due to the advancement in network technology. Modern technology plays a vital role in the field of mobile computing (Jin *et al*; 1999). It is important to see the factors which affect the performance of mobile database. There are many factors which affect the performance e.g. cache size. Greater the cache size, better the performance because more data can be loaded in cache, hence speed of CPU to access data will be faster. So, the size of cache also affects system performance. Replacement algorithms for data access play a vital role in mobile database performance. This research work emphasizes on the comparison of semantic intra file technique with semantic inter file and fully associative techniques. It is proved that semantic intra file technique is the best one along with a bulk of advantages of mobile database, it has several disadvantages like unreliability of connections, security and privacy threats etc. On the mobile device, a cache is used to save data that is frequently requested or used. Its purpose is to avoid from the loss of data due to connection failure. Several techniques have been proposed to improve performance of cache of a mobile device. But the problem is that which one of these techniques is more efficient and reliable, so that, manufacturers of the cache for mobile device may choose the best one. Therefore, there is a need to make such a study that may help and guide manufacturers of cache of a mobile device in making efficient and reliable cache and improving performance of mobile database.

MATERIALS AND METHODS

Semantic-based Caching: Among different file access techniques, semantic based cache techniques are different which use the inherent relationship between files to perform the functioning or hoarding (Geoffrey and Popek, 1997). Semantic intra file, does not use this relationship to prefetch files or save them, rather it focuses on its efforts on saving these relationships in the cache. An eviction or removal index was calculated based on these relations. In semantic intra file strong relations are saved but weak ones are omitted. This approach not only gives an effective hit ratio but also decreases the

communication overhead as compared to other replacement algorithms such as LRU(least recent use), LFU(least frequent use) and Greedy-dual size (Edith *et al*; 1998) when run against the distributed file system (James *et al*; 2005). Semantic intra file is thought to be important as in semantic cache, the relationship between files is considered but in conventional approaches it is mostly rejected or ignored. In semantic cache, the need for replacement algorithm is minimized which means the increased file availability, reliability and reduced synchronization overhead. User patience depends on importance of the used files and expected fetch delay. Importance of priority files is specified by user in some systems (Maria *et al*; 2002). In the present study, a thought has been developed to remove this overhead by establishing file relationships based on which caching is done. Semantic intra file provides better hit rates than LRU while in some cases, performing half the number of replacements. It is observed that by allowing more files to remain in the cache, the user can be given more options to use.

Caching Architecture: Fig. 1 shows the client server cache architecture. It shows how query is moved forward from client side to server side and server side to client side (David *et al*; 1992).

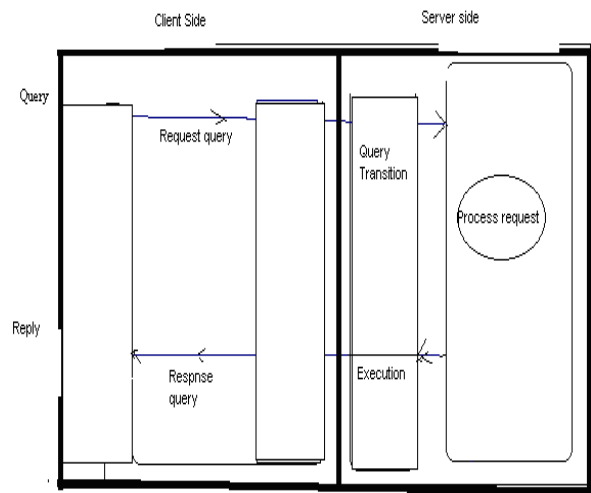


Fig-1: Client server Cache Architecture

In semantic cache inter file relationship, Brian reported that data or information is obtained from the distributed file system and it is found on priority basis. Some times, total file is checked by a very short procedure. It is observed that almost 50 to 60 % file access occurs only in one precursor (Brian *et al*, 1997). In semantic intra file technique, cache relationship is considered better where files are opened before they are closed. It is the concept of time sharing. Mostly, it is observed that time sharing concept is raised where there some priority has to be settled down i.e. which process is

occurring first. The concepts of time sharing are raised in the multiprocessing functions and in next generation like hyper threading it gives better result.

Semantic intra file relation is a basic technique to enhance the performance in cache.

Cache Mapping: When we want to know how we can enhance the performance of cache, there are two main factors which affect the performance i.e. hit ratio and searching speed. When processor finds data in cache, it maps the cache line with memory address. Data is said to be found if cache line is mapped with the memory address. Searching speed depends upon the region of cache in which searching is performed. Network speed also plays an important role in increasing the performance.

Semantic Caching Vs Fully Associative: This research focuses on the comparison between semantic based caching and fully associative mapping in distributed mobile environment. Semantic based caching is fast and efficient in mobile environment where communication disconnection, speed, bulk of data management and other geographical issues are present as a threat. Semantic intra file gives better result as compared to the previous technique like Fully Associative Mapping. The authors have briefly discussed all these techniques below and extracted some results with the comparative study of these techniques.

Fully Associative Mapping: If the replacement algorithm is free to hold any data in the cache, the cache is called fully associative. In the cache, a copy of a particular data from main memory moves to cache for processing and the replacement algorithms like LRU or MFU (most frequent use) are responsible for all its working.

Semantic inter file caching: Semantic inter file caching gives efficient solution for query processing in which data and its semantic information are cached on the client side in distributed environment. It establishes a relation

of one file to another file on the basis of semantic caching, fetches data at once and saves it in cache at the time of network failure

Semantic intra file caching: Semantic intra file caching gives best solution for query processing as compared to semantic inter file caching and fully associative mapping technique. It gives the inherent relation of a file in which data and its semantic information are cached in the client side.

RESULTS AND DISCUSSION

Simulator Working: In the designed software (simulator) semantic based techniques are used which give the best efficiency in data finding in the cache. These techniques are used to find the memory addresses and cache lines in cache which give the desired data or information. For next data finding, processor goes to the neighbor of previous address to find the data. This rule is called 90% to 10 % rule (Brian *et al*; 1997). It is observed that behavior of access pattern is totally changed. The performance of semantic cache would be changed (Bjorn *et al*; 2006) by changing cache size, speed, and replacement strategies. Greater storage space and using distributed file system are the main advantages of this semantic cache technique. In simulator (Fig. 2), the pane of 'Statistics' shows the hit ratio (Hit = 21) of access pattern data. It shows 70% to 80% hit ratio. The simulator's 'Statistics' pane also shows the miss ratio (Miss = 7), to perform the functioning or hoarding (Geoffrey and Popek, 1997). This actually does not measure the exact ratio. But here the simulator makes this fact clear that the semantic intra file technique is the best one. For semantic intra file relationship, data or information is obtained from the distributed file systems. Its own file traces activity is mostly on precursor. Sometimes, total file is checked in a very short time. It is observed that almost more than 50% file access is occurring in only one precursor (Brian *et al*, 1997). Fig. 2 shows the working of simulator.

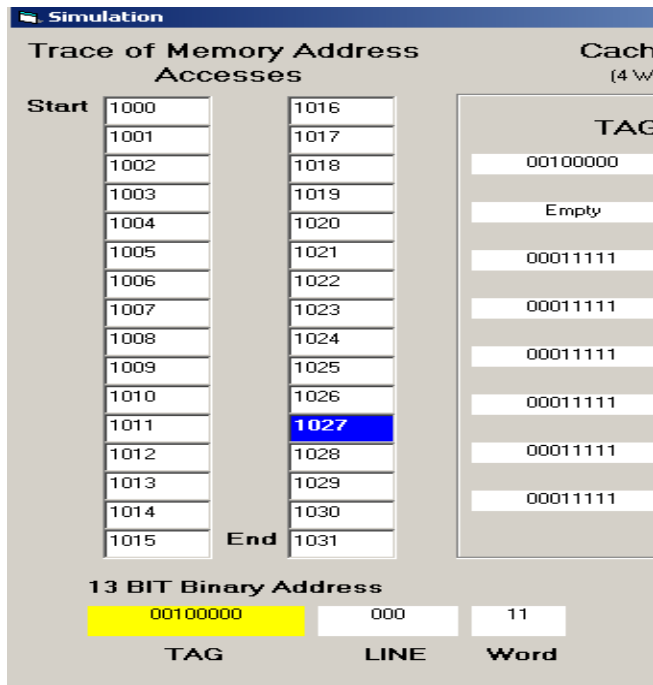


Fig-2: Cache simulator interface

Table-1: A comparative summary of the three techniques

Address Call	Semantic Intra file caching		Semantic Inter File caching		Fully Associative Mapping	
	Hit Data	Hit Ratio	Hit Data	Hit Ratio	Hit Data	Hit Ratio
15	10	67%	7	47%	3	20%
20	15	75%	11	53%	4	19%
25	19	74%	13	52%	5	20%
30	24	77%	20	67%	7	27%
35	29	79%	20	62%	10	30%
Average hit ratio	-	74.4%	-	56.2%	-	23.2%

Statistical Analysis of Fully Associative: In fully associative technique, simulator gives almost 20% to 30% hit ratio which is the maximum efficiency result for this technique. Figure 3 shows average hit ratio $(20+19+20+27+30)/5 = 23\%$ (Table 1). The results observed by (Shaul *et al*; 1994) were different from these results. Because here, it is compared with semantic cache techniques.

Statistical Analysis of Semantic Inter File: Semantic inter file cache technique shows the 50% to 70 % hit ratio (fig. 4) as $(47+53+52+67+62)/5 = 56\%$ (See table 1) (Sharun *et al*; 2004). It describes that semantic inter file

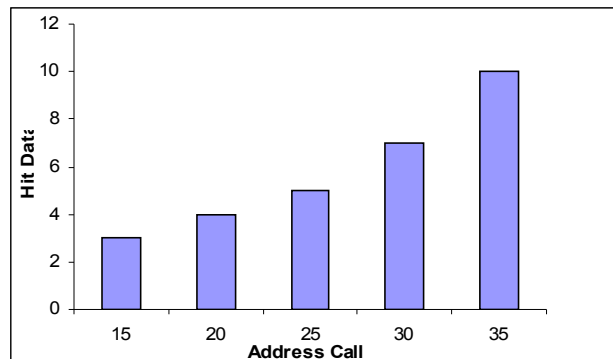


Fig-3: The average hit ratio of fully associative technique.

has more hit ratio as compared to fully associative technique.

Statistical Analysis of Semantic Intra File: Figure 5 is a graph of semantic intra file cache technique. It shows that as we increase more addresses, the hit ratio of the data also increases. At a particular point maximum hit ratio is 80%. Average hit ratio is almost $(67+75+74+77+79)/5 = 74.5\%$ to 75% (See table 1).

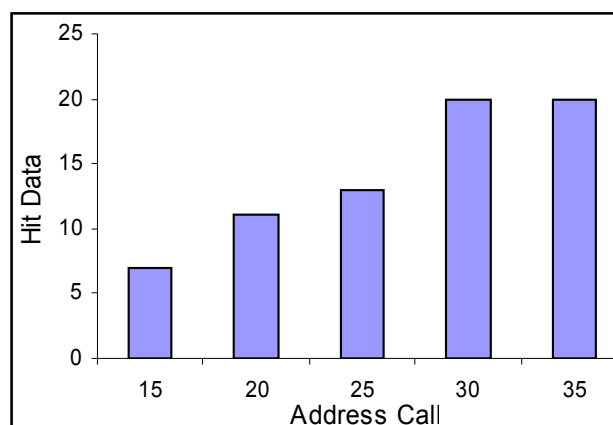


Fig-4: The average hit ratios of semantic inter file technique.

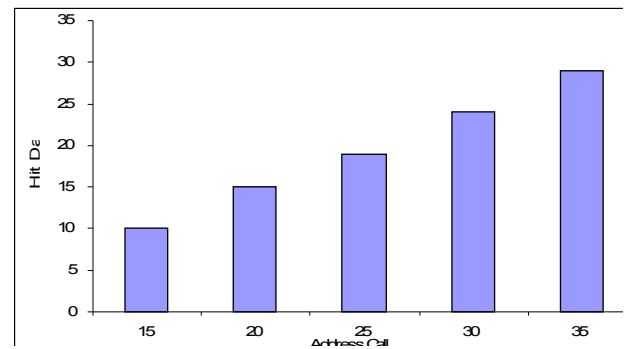


Fig-5: The average hit ratio of semantic intra file technique.

Comparison of these three techniques shows that semantic intra file is best, from performance point of view having almost 75% hit ratio. Semantic inter file (56% hit ratio) has less as compared to semantic intra file (75% hit ratio). Fully associative technique has least performance as compared to the above two techniques which has almost 23% hit ratio. Figure 6, a pie graph, shows comparison of the hit ratio between these three techniques which is measured under simulator results.

Pie Diagram of Semantic Inter File, Intra File and Fully Associative: Pie diagram (fig. 6) shows the comparison among these three techniques.

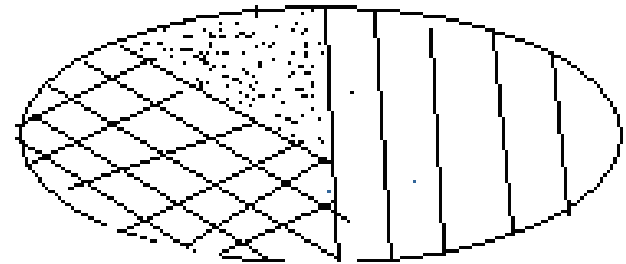


Fig-6: Comparison of the three techniques in form of pie graph.

Semantic intra file=Parallel lines area
Semantic inter file=Net lines area
Fully associative = Dotted area

Conclusion: In this study, the authors of this research work present a comparative study of the three techniques (fully associative, semantic inter file, and semantic intra file) and proved that semantic intra file technique is the best one. The results show that semantic intra file technique has better performance in the mobile database system. Hit ratio of fully associative, semantic inter file and semantic intra file techniques is 23%, 56% and 75% respectively.

Future Work: This research focuses on the semantic cache techniques and define a comparison between semantic inter file, intra file and fully associative techniques. For future extension, this area requires the

improvement in the semantic cache implementation techniques, such as grammar oriented searching algorithms, best search techniques, greedy algorithm and size of cache.

REFERENCES

- Bjorn, P. J., M. Arinbjarnar, B. Porsson, M.J., Franklin and D. Srivastava. *Performance and Overhead of Semantic Cache Management*, **ACM Transactions on Internet Technology (TOIT)** 6:302-331, ACM Press New York, NY, USA ISSN: 1533-5399 (2006).
- Brian, D. N., M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn and K. R. Walker. *Agile application-aware adaptation for mobility*, **ACM Symposium on Operating Systems Principles Proceedings of the sixteenth ACM symposium on Operating systems principles** Saint Malo, France: 276 - 287 ACM Press New York, NY, USA ISSN:0163-5980 (1997).
- David, J. D., M. J. Franklin, M. J. Zwilling, C. K. Tan and M. J. Carey. *Crash Recovery in Client-Server* EXODUS ACM SIGMOD International Conference on the Management of Data:165 – 174 ISBN:0-89791-521-6 San Diego California, USA. (1992).
- Darin, C., and J. F. Roddick. Context-Sensitive Mobile Database Summarisation, **Proceedings of the 26th Australasian computer science conference – 16**:139-149, (2003).
- Edith, C., B. Krishnamurthy and J. Rexford. *improving end to end performance of web using server volumes and proxy filters*, ACM SIGCOMM Computer Communication Review, Proceedings of the ACM SIGCOMM '98, 28:241-253 (1998).
- Geoffrey, H. K., and G. J. Popek. *Automated hoarding for mobile computers*, ACM SIGOPS Operating Systems **ACM Symposium on Operating Systems Principles Proceedings of the sixteenth ACM symposium on Operating systems principles** Saint Malo, France Pp. 264 - 275 ISBN:0-89791-916-5 (1997).
- Jin, J., A. S. Helal and A. Elmagarmid. Client-server computing in mobile environments, **ACM Computing Surveys (CSUR)**, 31:117-157 (1999).
- James, L. L., G. Santhanakrishnan, A. Amer and, P. K. Chrysanthis. *Self-Tuning Energy-safe Predictors*, **International Conference On Mobile Data Management Proceedings of the 6th international conference on Mobile data management** Ayia Napa, Cyprus SESSION: Caching and replication pp. 125 - 133 ISBN:1-59593-041-8 (2005).
- Maria, R. E., B. E. John and M. Satyanarayanan. *The importance of translucent cache in mobile computing system*, **ACM Transactions on Computer-Human Interaction (TOCHI)** 9: 42 – 67 Publisher ACM NY USA ISSN:1073-0516 (2002).
- Sharun, S., W. Shi, and H. Lufei. *An Adaptive Distributed File System for Heterogeneous Network Environments*, **Proceedings of the Parallel and Distributed Systems, Tenth International Conference on (ICPADS'04) - 00**:145 , IEEE Computer Society Washington, DC, USA , ISBN~ISSN:1521-9097 , 0-7695-2152-5 (2004).
- Shaul, D., and R. Ramakrishnan. *A performance study of transitive closure algorithms*, **Proceedings of the ACM SIGMOD international conference on Management of data**, Pp:454-465 Minneapolis, Minnesota, United States (1994).