A Survey on Applications of Genetic Algorithms and Fuzzy Logic in Caching

Mahip M.Bartere*, Dr. Prashant V.Ingole**
*(Asst.Professor Department of Computer Science & Engineering
G.H Raisoni College of Engineering & Management, Amravati)
** (Principal, G.H Raisoni College of Engineering & Management, Amravati)

ABSTRACT
Caching performance can be improved by designing good replacement policies. This paper,discuss the various approaches that were designed based on genetic algorithms and fuzzy logic to optimize the performance of caching. The approaches discussed here proved to be more effective in solving the problems as compared to the conventional techniques that were used earlier in this problem domain.

Keywords - **Genetic algorithms, Cache replacement, Fuzzy logic.**

I. INTRODUCTION

The storage capacity managed by cache is of limited size, so it requires removal of old objects from the cache whenever new objects are brought in. The decision regarding which objects to remove from the cache is made by designing a suitable cache-replacement algorithm, which has been under continuous research. The performance of cache could be improved significantly by implementing genetic fuzzy algorithm techniques

II. A SURVEY FOR GENETIC ALGORITHM TECHNIQUES

Athena Vakali [5] presented a model that introduced the concept of applying genetic algorithms and evolutionary programming to cache replacement process. The proposed models adapted the idea of evolutionary computation in order to preserve a consistent cache "population" of information objects. The aim was to improve the cache population in terms of its reliability and accessibility. For better utilization of the cache area, genetic algorithm (GA) approach was considered for the cache replacement process. The reasons for considering the genetic algorithms were as follows: a) cache should contain the fittest (i.e., non-stale, frequently accessed) information objects and b) cache content that had a large number of information objects (stored files) required optimization.

The cached objects were modeled as individuals considered for evolution. Genetic operators such as crossover and mutation were applied to selected cache objects for creating stronger cache generations. Evolutionary programming (EP) techniques were similar to GAs, but they placed emphasis on the behavioral linkage between cached objects chosen as "parents" and their offspring. EP was considered useful in optimization problems and hence applied to cache replacement.

The cache hit ratio and byte-hit ratio were the performance metrics considered in the model. The crossover and mutation probability values were $P_{\text{crossover}} = 0.6$ and $P_{\text{mutation}} = 0.001$. The approaches outperformed the conventional least-recently-used (LRU) policy adopted by the most currently available proxies. Significant improvements in cache hit ratio and byte-hit ratios could be achieved by providing proper objective function to the evolution process.

III. A Survey for Fuzzy Algorithm Techniques

Fuzzy logic was an extension of Boolean logic, which dealt with partial truth concept denoting the extent to which a proposition was true. The general fuzzy logic controller or fuzzy inference system (FIS) consists of three functional blocks: fuzzifier (input stage), defuzzifier (output stage) and an inference engine (processing stage) containing a fuzzy rule base. *Fuzzification* - it changes the crisp input values into fuzzy input sets.

***Inference Engine* - it makes conclusions using the rule base.**

Rule Base - it is a control knowledge-base characterized by a set of linguistic statements in the form of "if-then" rules that describe a fuzzy logic relationship between inputs and outputs.
Defuzzification - it allows to change the fuzzy value obtained after inference processing into crisp output value that represents the final decision.
The number of rules has a direct effect on the time complexity of fuzzy inference systems. Performance of the system can be improved by having fewer rules. The characterization of fuzzy system operation depends on the proper choice of input and output variables. The fuzzy algorithm has the ability to easily adapt to the characteristics of the workload.

IV. Cache Replacement by Fuzzy logic

The frequency and recency were the two important parameters considered for cache replacement decisions. Even though correlation exists between recency and

frequency, it is not the same for all kinds of workloads. It is impossible to derive an exact mathematical formula to describe this relation. The simple method to achieve this is to model the parameters through fuzzy logic.
Giacomo Valli *et al.* [2] proposed an algorithm that applied a set of fuzzy control rules to identify the pages for eviction from the cache. The algorithm considered characteristics and properties of workloads and then applied qualitative reasoning to identify pages to evict from the cache. To represent the process state, three input variables were chosen. The variables described page in terms of its size, access frequency and access recency. The output variable represented the probability of replacement for each page. Fuzzy sets were defined for these variables, with the membership functions describing the degree of membership of the variable in fuzzy set. The variables "size" and "frequency" have three membership functions (low, medium and high), and the variable "time" has five membership functions (very low, low, medium, high and very high). The fuzzy "and" operator was used to evaluate the antecedent and compute the degree of truth for each rule.The "maximum" operator was applied by the aggregation process to combine the outputs. Defuzzification used in the algorithm was based on the centroid method.

Two sets of fuzzy rules (Fuzzy 20, Fuzzy 12) were defined, and their performance was tested against LRU, LFU replacement algorithms. The efficiency of replacement policies was evaluated as a function of the cache size. Fuzzy 20 was aimed at maximizing the hit rate (HR), whereas Fuzzy 12 was aimed at maximizing both the hit rate and byte hit rate (BHR). Fuzzy 20 and Fuzzy 12 achieved an HR of more than 90% with a cache size as low as 5% of its capacity. The fuzzy algorithm performance was at least as good as the performance of the other algorithm (like GDS algorithm).

Mojtaba Sabeghi *et al.* [1] proposed a fuzzy algorithm for cache replacement, which treats decision parameters as fuzzy variables. The algorithm was confined to replacement of uniform cache objects with fuzzy logic. The most common shape of a membership function used in the fuzzy approach is triangular. In the proposed model, the input stage consisted of three linguistic variables: a) recency, which represents spatial locality of references; b) frequency, which represents temporal locality of references; and c) reuse distance, which is the distance between two consecutive references to the page. Frequency has more impact for larger caches; and recency, for smaller cache sizes. Reuse distance enriches the system to become suitable for weak locality workloads. The algorithm steps are as follows:

1. Inference engine takes as input the recency,frequency and reuse distance of each used page 'P' in the cache.
2. Output of the inference engine is considered as "swap priority" of page P.
3. Page that has the highest swap priority will be removed from the cache.

The algorithm was tested with trace-driven simulations considering various types of workloads. The experimental results reflect that the fuzzy approach is suitable for looping, probabilistic and temporal patterns of reference, and it performs better in mixed reference patterns.

VI Conclusion

In this paper, we have discussed various approaches proposed by authors to improve the performance of caching using neural networks, genetic algorithm and fuzzy logic. In case of cache replacement, the basic inputs considered by fuzzy are frequency and recency. In addition to that, the techniques also considered size and reuse distance as parameters for replacement process. The techniques produced improved results over the conventional LRU, LFU algorithms. Genetic algorithms and evolutionary programming techniques were considered to solve problems in cache replacement. The techniques proved to be effective in providing solutions and improving the performance of the system compared to conventional approaches.

Since each technique required different inputs and the experiments were carried out with different setups, it was difficult to compare the performance of different techniques at this level. It would have been better if the experiments for these techniques had been conducted with similar inputs and similar experimental setups, and the results then compared. The techniques could well be considered for other issues that are related to caching , such as cache consistency, quality of service and policy management. Further work should experiment with evolutionary schemes such as evolutionary strategies, simulated annealing and threshold acceptance.

REFERENCES

[1] M. Sabeghi, and M. H. Yaghmaee, "Using Fuzzy Logic to Improve Cache Replacement Decisions", *International Journal of Computer Science and Network Security* , vol. 6, no. 3A, Mar. 2006.
[2] G. Valli, and M.C.Calzarossa, "A Fuzzy Algorithm for Web Caching", proceedings of *International Symposium on Performance Evaluation of Computer and Telecommunication Systems* (*SPECTS*), pp. 630-7, SCS Press, 2003.
[3] D. E. Goldberg, “Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking” *Complex Systems,* 5 (1991) 139-167
[4] A. Silberschatz, and P. B. Galvin, *Operating System Concepts* (1998) 300-313
[5] 23.1)A. Vakali, "Evolutionary Techniques for Caching", Distributed and Parallel Databases, *Springer* , vol. 11, pp. 93-116, 2002.
[6] Koza, John R., Bennett III, Forrest H, Andre, David, and Keane, Martin A. Genetic Programming III: Darwinian Invention and Problem Solving (1999)