iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** IV **Month of publication:** April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4684

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

A Survey on Cache Replacement Policy

Nishtha[1], Komal Agrahari[2]
[1, 2] *Department of Computer Science and Engineering, Madan Mohan Malaviya University of Technology, Gorakhpur*

Abstract: Caching techniques can enhance the performance of a system by using the substitution strategy called as replacement policy. The main focus of this survey paper is learning about several replacement algorithms with their advantages and disadvantages. In this paper we studied various replacement policies and compare their performance based on miss rate, hit rate.
Keywords: Miss rate, hit rate, cache line, replacement policy, cache memory.

I. INTRODUCTION

Cache memory is small and fast memory which needs copies recently accessed instruction and data.it is the memory which is very nearest to the CPU, all recent instruction are stored into cache memory. Cache memory is attached for storing the input which is given by the user and which is necessary for the CPU to perform a task. The capacity of cache memory is a smaller amount but this is a very costly memory. Cache memory is faster as compared with main memory. But data and instructions are stored in the form of memory hierarchy. So the speeds of memories are arranged in high to low in memory hierarchy. Computer memory should be fast, large and expensive. The main memory is to obtain the highest possible access speed while minimizing the total cost of memory system. First of all the CPU will check the information is available in the cache .that is called hit rate. If information is not found in the cache it's called miss rate occur. With the help of replacement algorithm is for reducing miss rate and increasing the hit rate of the cache memory. when cache memory turn out to be full it means there are no space in cache memory and another new block is ready to insert in cache memory. So replacement policy is using for this purpose. In this paper we elaborate the LRU replacement policy. In this policy where we try to remove the page which has been used old page which has not been used for the longest time in main memory on which will be selected for replacement .it is like optimal page replacement algorithm but looking backward in time. LRU replacement remove page least recently accessed until there is enough space for new block. But it's slightly better than FIFO replacement algorithm. The CPU will check word in the cache if word is available and then send a required word to CPU. when word is not available in the cache at this condition main memory have words which is equivalent to required word .then cache read the word and send to the CPU. This called is locality of reference. The localities of reference are three types that is one are temporal locality reference, sequential locality reference and spatial locality reference. The CPU has different levels of cache. The cache levels are l1, l2 and l3.the data are arranged as a hierarchy of cache levels. In cache level 1 the speed of level L1is greater than L2and L3.but the size of L1 is less than L2and L3.while level l1 has lowest storage. It's made into the processor chip.l1 is also called internal and primary cache. In cache level 2 the speed is greater than l3 but smaller than l1.l2 has more storage capacity .it built directly into the separate chip. In the cache level L3 is built on the motherboard but it's separated from processor chip. The speed of L3 is slower than L1 and L2cache. But the size of l3 is greater than L1 and L2 [1].

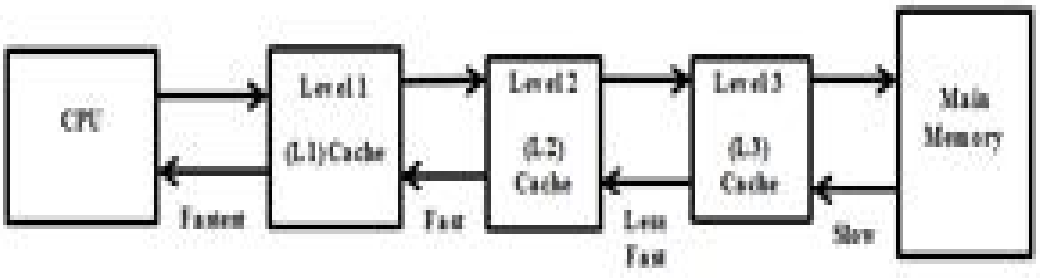


Fig.1 Three level cache

II. PARAMETER OF CACHE MEMORY

The parameter of cache memory as follows [4].

A. Cache hit: A reference item is found in the cache by the processor.
B. Cache miss: A reference item is not present in the cache.
C. Hit ratio: (number of bits*100) / Total number of bus cycle

1) Hit ratio is calculate the number of hits (success in finding the word in cache)
2) Bus cycle (total CPU reference to memory), (number of hits +number of miss) done by CPU.

D. *Miss penalty*: Additional cycle required to serve the miss.
E. *Latency*: time to retrieve the first word of the block.
F. *Bandwidth*: time to retrieve the rest of this block.

III. RELATED WORK

In this part of this paper we discuss many cache replacement strategies. This survey explains the result and simulation of each replacement strategy. Here we define the overview of each presented replacement strategy [3**].**

A. Optimal Replacement Based on The Future Reference

In this replacement strategy select victim cache line on the basis future reference. The optimal replacement strategy is ideal for all type of replacement algorithm. This strategy is not useful for dynamic system even future reference known or not known. If future reference is known in that condition computational cost create problem with the help of this replacement policy we calculate the lower limit of the cache misses number. This policy is useful for compare of all the replacement policies in term of performance.

B. Random Replacement

In this type of replacement strategy select the cache line randomly available cache lines. All the cache lines have equal probability for eviction in replacement .in this replacement minimum compatibility of the implementation process as compare to other replacement policies. Because of here not guarantee of steady hit rate ratio. Here cache arbitrary remove through cache line. In this type of replacement policies are all over cost about the hardware equipment for implementation. In this replacement policy cache line is not selected on the basis of past references and not requires any storage memory. The Random replacement is not suitable for high performance it's not give the steady hit rate .this strategy is famous in the embedded application due to limitation of hardware.

C. Arrival Replacement Strategy Based on Arrival Time

This type of replacement policy gives the predictable cache behaviour in under the lowest complexity. This policy works according to first in first out. according to FIFO select the oldest cache line as like as victim cache line as like as victim cache line in the set for replacement when we select the cache line than we store this cache line in the register for storing arrival time. For cache line we use the per bit register for storing the arrival time. For cache misses stored in the register in cache line and compared with victim cache line. When we use the alternate implementation then we maintain the FIFO queue on the basis of arrival time on blocks of cache. When replacement is require in the cache then bloc is removed from the head of the queue. Circuitry of replacement is modified and activated for cache miss. This strategy is not critical path. This is suitable for embedded application where memory reference related to the FIFO order.

D. Frequency Replacement Strategy Based on Reference Count

In this type of replacement select the victim cache line according to number of references. Frequency replacement strategy is used for the estimate the block which is reference according to probability some common strategies in under the Frequency replacement strategy such as Least frequently used –Dynamic Aging (LFUDA) and Least Frequent used (LFU).

1) *Least Frequency Used Replacement Strategy*: LFU substitution system chooses the target cache line derived from the recurrence of access of the cache lines in the set. LFU needs a frequency computation register for each cache line to keep up the quantity of references. The value of frequency register value of any block is set to 0 if any block is copied from main memory and this value is changed to 1 whenever this block is referenced. The cache line with the least frequency is considered as the victim cache line by LFU as it has the lowest frequency. Frequency count register is refreshed after each cache access which results in increase in cache access time. Any polluted cache line remains in the cache for long duration resulting in polluting others also. One such execution is LFUDA.

2) *Least Frequency Used – Dynamic Aging Replacement Strategy LFUDA*: LFUDA replacement action disables the cache pollution botheration of LFU replacement strategy. This is done by using a reference calculation register for calculating age. For each cache access reference count register is incremented. When this reference count register value reaches to threshold specified by algorithm, the frequency of cache lines is decreased. In order to store aging values an additional register is used called reference count register. Like frequency county register, this reference count register is updated for each cache access which results in increasing cache access time and power consumption. It as well needs added circuitry for appropriate shifting of frequency count registers. This will increment the complication, power consumption and admission time of this action more.

E. Recency –Reference Time Based Replacement Strategy

This policy is founded on reference time used for finding a target cache line, recency based cache replacement Policies are divided into two types:

1) Most Recently Referenced Replacement Strategies: The generally utilized substitution methodologies in most recently referenced experiential are Most Recently Used replacement strategy (MRU) and Priority Cache Replacement strategy (PC). The MRU substitution system picks the most recently utilized cache line from the set for exile. Due its bad temporal locality this strategy is not popular. With every cache line PC links a data priority bit. The compiler, through two extra bits related with every memory access instruction, allots priorities. These two bits display whether the information priority bit ought to be set as low or high. The cache line with the most minimal priority is the one to be replaced. Added hardware requisite in the usage of register operated to store the recency data. The size of the register used for storing MRU data is log2N where N is the associativity of the cache. The most recently referenced systems are more difficult to execute when contrasted with arbitrary, FIFO and frequency based methodologies because of the fact that for each cache accesses it modifies its own particular register as well as alters all the other enroll register in the set.

2) Least Recently Referenced Replacement Strategies: In this type of replacement strategy is to select a victim cache line that has not been used for longest time. This algorithm is vast replacement algorithm. In Least Recently Replacement Strategy many modification policies are Modified LRU (MLRU), Least Recently Used, Software assisted LRU (SALRU), Pseudo LRU (PLRU), Late LRU (LLRU), Early Eviction (ELRU), and Cache System with replacement control (cache/RC). These Replacement strategies are modification of LRU

3) Least Recently Used (LRU): It is page replacement policy algorithm where we try to remove page least recently access. The efficiency of Least recently used (LRU) reasons either decrease or same number of interrupts .it is slightly better than FIFO replacement algorithm. And the implementations of LRU replacement algorithm are counter and stack. This method may be without problems carried out in the two way set associative cache enterprise. It's complicated to implement because of this LRU replacement algorithm looks for data in past. But this policy like optimal replacement algorithm. The advantage is good to approximate MIN .In figure show the least recently used policy in the form of tree [2].

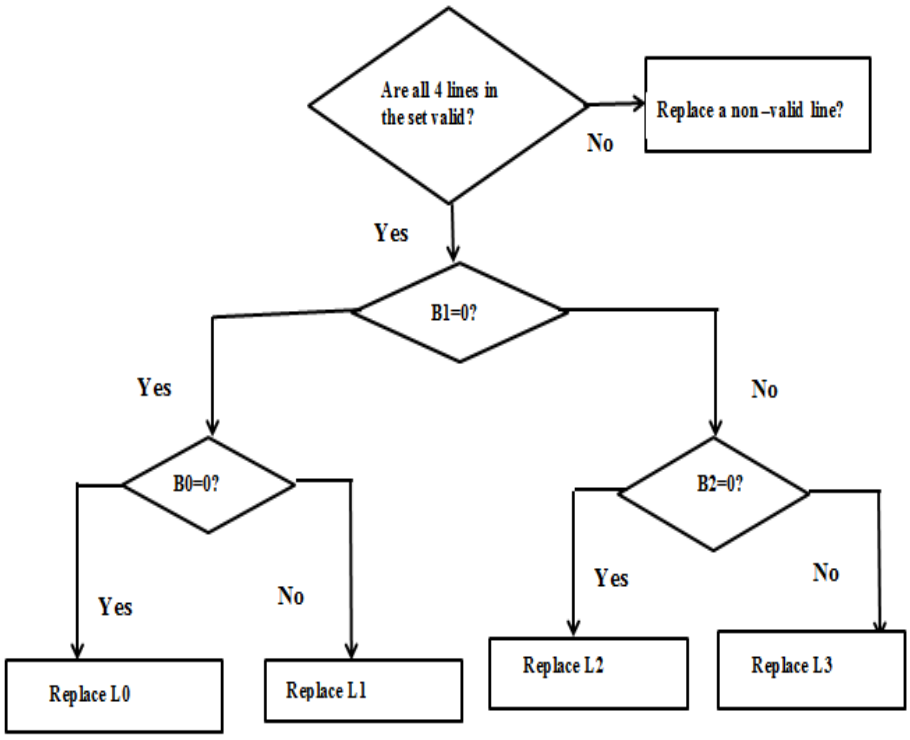


Fig.2 Least Recently Used Policy in the form of tree

Table I Complexity Comparison Table

Heuristic	Storage Requirements [bits]	Action on cache hit	Action on cache miss
Random	Log_2(ways)	None	Update LFSR register
FIFO	nsets.log_2(ways)	None	Increment FIFO counter
LRU	Nsets.way.log_2(ways)	Update the LRU stack	Update the LRU stack
PLRUm	Nsets.ways	Update the MRU Bit(s)	Update the MRU bit(s)
PLRUt	Nsets.(ways-1)	Update the tree bit(s)	Update the tree bit(s)

In TABLE I discussion about all replacement algorithm basis on the increase hit ratio and reduce the miss rate [2**].**

F. Recency + Frequency –Combination Reference Count and time of Last Reference Based Replacement Strategy

This algorithm is based on time and number of references. To select a victim cache line with checking the time and number of references in last reference. In this algorithm two replacement algorithm are merged but to give the good performance to any another strategies. This algorithm work better and outstanding performance. In another proposed replacement strategy to merged the LFU and LRU. Define the algorithm Second Chance Frequency –Least recently used. In which uses reference with counter to the cache line. But this algorithm creates a Recency- frequency control value (RFCV). That's combined with Frequency and Recency. If the counter value is increment the set to 1 that's a cache hit and then go to ready for entry new block in cache. And in the case of the LRU cache line smaller than RFCV. So the cache lines in LRU state that will become Second chance one.

IV. CONCLUSIONS

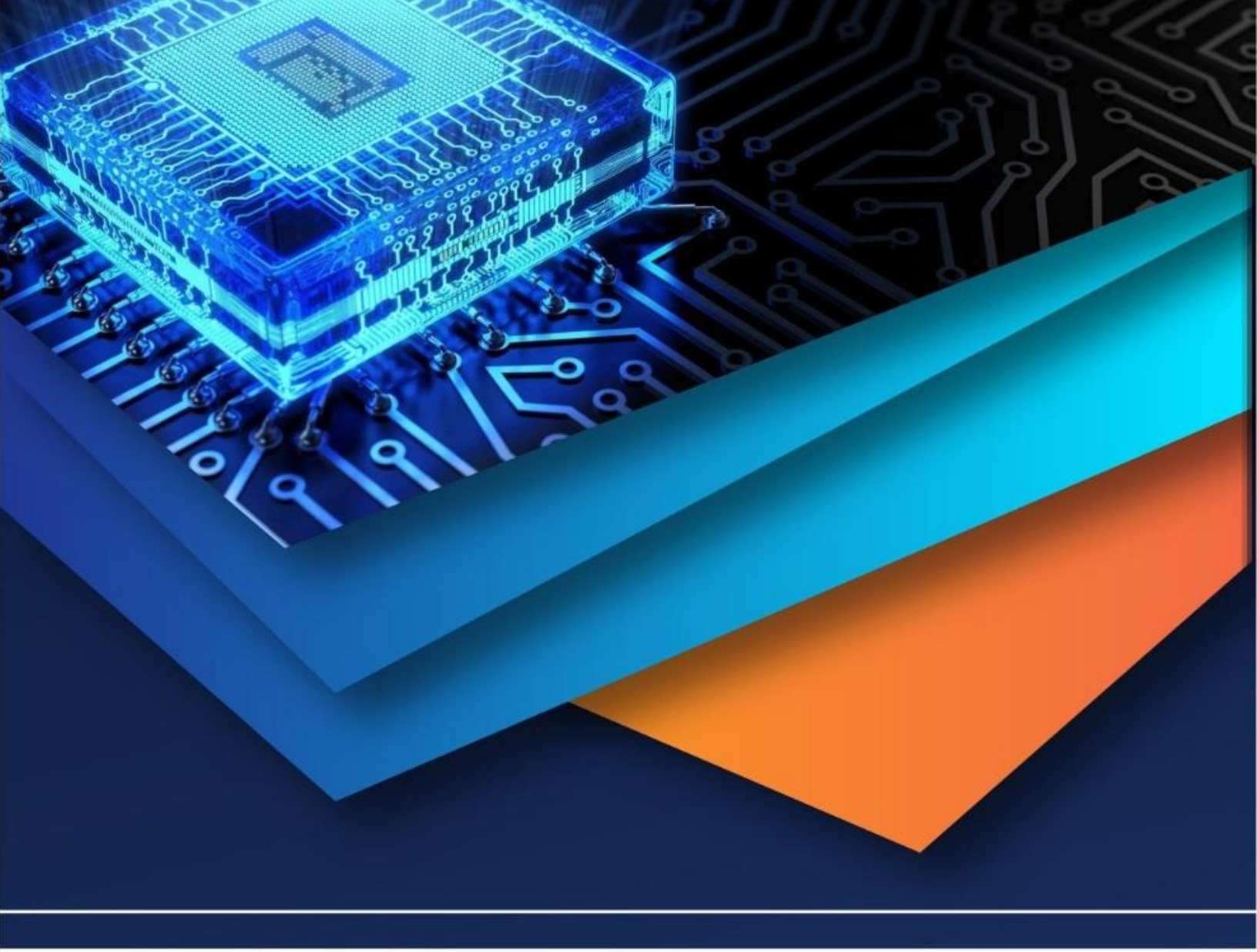
The survey of this paper we elaborate the replacement algorithm. We studied on various replacement algorithms such as LRU, FIFO, and Random and optimal, Most recently used. After analysis we found that the best accessible replacement algorithm is LRU. Because of to reduce the complexity of the processor and to decrease the cache misses rate for the overall performance of cache memory.

V. ACKNOWLEDGMENT

I am grateful to thanks my guide Dr. P.K. Singh and senior teacher Komal Agrahari for valuable support, encouragement and inspiration. Without guidance I cannot complete of my project. I like to say thank you my parents who always support and encourage me. I am deeply indebted to my parents, friends and my guide.

REFERENCES

[1] W. Stallings, "Computer Organisation and Architecture," Eighth Edition, 2011
[2] Hussein Al-Zoubi, Aleksandar Milenkovic, and Milena Milenkovic, "Performance evaluation of cache replacement policies for the SPEC CPU2000 benchmark suite," In Proceedings of the 42nd annual Southeast regional conference ACM , pp. 267-272, 20
[3] Parag Panda, Geeta Patil, and Biju Raveendran, "A survey on replacement strategies in cache memory for embedded systems," In Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), IEEE, pp. 12-17, 2016
[4] Arora, Kapil, and Dhawaleswar Rao Ch, "Web Cache Page Replacement by Using LRU and LFU Algorithms with Hit Ratio: A Case Unification", (IJCSIT) International Journal of Computer Science and Information Technologies 5, no. 3 (2014): 3232-323
[5] Swadhesh Kumar, and P. K. Singh, "An overview of modern cache memory and performance analysis of replacement policies," In Engineering and Technology (ICETECH),IEEE International Conference, pp. 210-214, 20
[6] Manish Kumar Verma, and P. K. Singh, "Enhanced Approach for Cache Behaviour and Performance Evaluation," International Journal of Advanced Research in Computer Science volume 8, no. 3, 201Komal Agrahari, and P. K. Singh, "Realisation of Cache Optimisation using New Technique," International Journal of Advanced Research in Computer Science volume 8, no. 3 , 2017 .



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY