DOI: http://dx.doi.org/10.26483/ijarcs.v9i3.5952

ISSN No. 0976-5697

Volume 9, No. 3, May-June 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

ENHANCED PERFORMANCE OF CACHE MEMORY

Nishtha
Dept. of Computer Science and Engineering
Madan Mohan Malaviya University of Technology
Gorakhpur,India

Komal Agrahari
Dept. of Computer Science and Engineering
Madan Mohan Malaviya University of Technology
Gorakhpur

Abstract: The focus of this paper is to understand the cache overall performance using the new strategy of cache replacement. We proposed a new strategy in which we use Hybrid LRU replacement policy. This proposed model is for hit rate and reducing miss rate. We study on replacement policy that is executed when one of the following occur no. of misses rate and no free cache line in physical memory .so degrade the system of performance that's by no free cache line in memory. The new techniques of cache replacement provide the maximal hit ratio as compared to existing policy.

Keywords: Miss rate; Hit rate; Replacement policy; LRU replacement policy; optimal replacement policy

I. INTRODUCTION

The focal point of this paper is to understand the cache overall performance using the new strategy of cache replacement policy. Within the previous work FIFO replacement algorithm and LRU replacement algorithm is used inside the sub blocks of cache memory to enhance the overall performance of cache. Now after studying lots of end result we have achieved that the proposed coverage has decreased quantity of misses compared to present policy. And we understand that miss rate has very essential function within the overall performance of the cache [1]. This paper report the improvement of cache overall performance by using optimization techniques. Cache memory is volatile, small sized and high speed memory .The frequently accessed and temporally data is stored in the cache. The located between CPU and main memory. the cache performance improve through the increase of hit ratio, decrease the miss rate and memory retrieved time and miss panelty.when CPU is want to a word than CPU is read a word from main memory [3]. First of all it will check word in cache memory and the word is available in the cache than send a required word to CPU. If word is not available in the cache .At this condition main memory has words which is equivalent to required word. Then cache read the word and sends to CPU. This is locality of reference. Three type of the locality of reference [2]. There are temporal locality of reference, Sequential locality of reference and spatial locality of reference. In the case of miss rate if no empty block in the cache to using replacement policy algorithm [5].

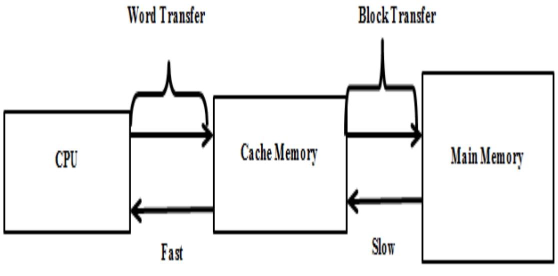

Figure1. Single Cache (Cache and main Memory)

II. CACHE REPLACEMENT

When cache misses are perform the existing line then calculate the new line. Existing line is replaced by new calculated line. Cache replacement produce lowest miss rate. This is the most point of this cache replacement algorithm. The cache replacement policies are commonly used to which data call in cache memory. Cache replacement policy [1] is not depending on how much data in the cache. There are many type of replacement policy algorithm as a LRU, FIFO, Random and MRU.

A. First in first out Replacement policy

The first page replacement algorithm is the FIFO .this is very simple to implement .the oldest page in the memory will be replaced for replacement .And the efficiency of FIFO replacement algorithm is not so good. More memory does not lead to better for FIFO anomaly [5]. This algorithm makes use of identical idea that stacks implementation usages in the microprocessor.

B. Least Recently Used Replacement Policy

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 [4] 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 [6].

C. Optimal Page Replacement Policy

Replace the page that will not be used for the longest. The miss rate of optimal replacement algorithm is lowest. The efficiency of this algorithm is highest .Optimal replacement policy is best algorithm than FIFO replacement algorithm and LRU page replacement policy .one end result of the invention of Belady's anomaly became the look for an OPT replacement algorithm. This algorithm is decreased miss rate as compared to all algorithms and it will disagree from Belady's algorithm [6].

© 2015-19, IJARCS All Rights Reserved

D. *Most Recently Used Replacement Policy*

Remove the data with we have used just most recently used .The data block currently use is replaced in this algorithm. Most recently replacement makes better performance when use data block in a cache. MRU replacement algorithm have more hit ratio than LRU policy .Due to their recall older data .

E. *Random Page Replacement Policy*

With the help of this algorithm we can remove the randomly chosen data. There may be no unique standards for replacement of some block. The present blocks are changed randomly[6].

F. *Second Chance Page Replacement Policy*

We take a reference bit for each data block in the memory. If initially the reference bit is set to 0. That is to pick out for replacement. Reference bit is already in the memory because of bit was referenced again and reference bit is set to 1.the reference bit move to the next block. While a block gets a next chance, there is an empty reference bit and this one point in time is changed to the present time. Therefore that second chance data block won't get replaced otherwise given second chance. Additionally if a block is accessed regularly sufficient to hold its reference bit set .It'll in no way be replaced. Second chance replacement algorithm is variance of FIFO replacement algorithm .This is fare better performance than FIFO replacement algorithm [2].

III. PROPOSED WORK

In this paper we proposed the method of Hybrid LRU replacement algorithm. So as the result is shown the enhance performance of cache memory through reduce of miss rate as compared to new policy.

The steps of Hybrid LRU replacement algorithm follow as

```
        •  CPU generates memory reference.
        •  Enter the reference with counter value
  {
  If (cache hit) {
  Counter = Counter+1;
  }
  If (Cache misses) {
  Enter the reference with counter;
  }
  Until Cache full;
  }
     •  If (Cache full && Cache miss)
  {
        •  Check future memory reference generated
     by CPU and Cache memory reference.
        •  If ( any of them will be matched )
  {
 Go for Optimal replacement policy
}
Else
{
Go for LRU replacement policy
}
```

Table I. Comparison between LRU and Hybrid LRU

No. of sets	*LRU (Hit Rate)*	*Hybrid LRU (Hit Rate)*
S1	4	5
S2	2	5
S3	3	4
S4	1	4
S5	0	2
S6	1	3
S7	3	5
S8	6	9

A. *Graphical Represantation of Proposed work*

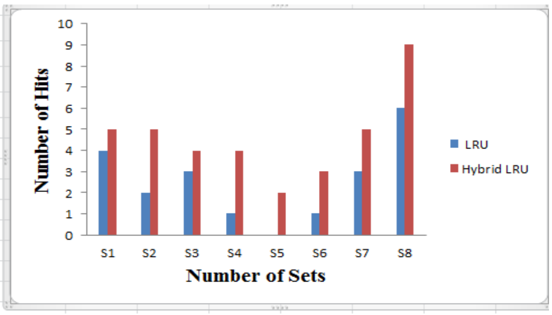

Figure 2. Graph of Comparison between LRU and Hybrid LRU

IV. CONCLUSION

The conclusion of this paper we proposed Hybrid LRU replacement algorithm. This algorithm is using for the hit ratio and reduce miss rate. Because of this method make more reliable execution of the cache memory and we get better performance of cache memory .with the help of this technique we used can increase the overall performance of cache memory. From the above table we found that the existing policy (LRU) has lower performance than our proposed policy (Hybrid LRU).

V. ACKNOWLEDGEMENT

I am eternally glad to thank you my guide Dr. P.K. Singh and senior Komal Agrahari for prized time, provision, encouragement and inspiration. Deprived of guidance I cannot complete of my project. I would likely to say thank you my parents who always provision and inspire me. I am sincerely grateful to my parents, my guide and friends.

VI. REFERENCES

[1] Mainak Chaudhuri, "Pseudo-LIFO: The Foundation of a New Family of Replacement Policies for Last-level Cache," Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, 2009, pp. 401-412.

[2] W. Stallings, "Computer Organization and Architecture," Eighth Edition, 2011.

[3] N. Beckmann and D. Sanchez, "Modeling cache performance beyond LRU," IEEE International Symposium on High Performance Computer Architecture (HPCA), Barcelona, 2016, pp.225-236.doi: 10.1109/HPCA.2016.7446067.

© 2015-19, IJARCS All Rights Reserved

[4] Swadhesh Kumar and P. K. Singh, “An overview of modern cache memory and performance analysis of replacement policies,” IEEE International Conference on Engineering and Technology (ICETECH), Coimbatore, 2016, pp.210-214, doi:10.1109/ICETECH.2016.7569 243.

[5] Manish Kumar Verma and P. K. Singh, “Enhanced Approach for Cache Behaviour and Performance Evaluation,” International Journal of Advanced Research in Computer Science, vol. 8, no.3, 2017, pp. 746-749.

[6] Komal Agrahari and P. K. Singh, “Realisation of Cache Optimization using New Technique,” International Journal of Advanced Research in Computer Science, vol. 8,no.3, 2017, pp. 750-752.

© 2015-19, IJARCS All Rights Reserved