IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, **p**-ISSN: 2278-8719
Vol. 3, Issue 10 (October. 2013), ||V5|| PP 21-25

A Two Layered approach to perform Effective Page Replacement

Khusbu Rohilla
[1](*Student of M.Tech CSE, GJU S&T, Hisar*

Abstract: - In a multi processor system, when the request are performed more frequently, more effective management of cache is required. Cache is small size memory available in system that holds both data and instruction. In case of shared cache, the management of cache is more critical. Cache memory provides the static storage to the processor for most required data and instruction. As of limited size, the contents of cache are been replaced regularly. To improve the efficiency of the system, an effective cache replacement algorithm is required. The presented work is in same direction. In this work a two level cache is been defined. In first level the prioritization of the cache contents is been defined based on the type of contents. At the second level the replacement is been performed by analyzing the page utility and the frequency of the access. The analysis of the work is been performed with different cache size. The analysis of work is been performed in terms of hit ratio and miss ratio. The obtained results show that the presented work has improved the efficiency and reliability of the multiprocessor systems.

Keywords: - *Cache, Frequency Analysis, Hit Ratio, Miss Ratio, Utility Analysis.*

I. INTRODUCTION

In most organizations, computing resources are underutilized. Most desktop machines are busy less than 25% of the time and even the server machines can often be fairly idle. Multiprocessor computing provides a framework for exploiting these underutilized resources and thus has the possibility of substantially increasing the efficiency of resource usage. The easiest use of Multiprocessor computing would be to run an existing application on several machines. The machine on which the application is normally run might be unusually busy, the execution of the task would be delayed. Multiprocessor Computing should enable the job in question to be run on an idle machine elsewhere on the network. From this point of view, Multiprocessor Computing looks like a great answer to numerous problems without prodigious management requirements. Convenience does not always imply simplicity and Multiprocessor enabled applications have to meet two prerequisites: first, the application must be executable remotely and with low additional operating cost; second, the remote machine must meet any special hardware, software, or resource requirements needed to run the application. Hence, remote computing is not only a matter of sending jobs to distant machines but also choosing the correct machine that has all the required software and hardware. Even though Computing Power (CPU) can be assumed to be the most commonly shared resource, it belongs to a large set of underutilized resources that are to be shared on the network such as storage capacity, software, internet connection and the like which also need to be considered.
From the scheduling viewpoint, each resource provider is modeled with three parameters: capability, job queue, and unit price. Capability is the computational speed of the underlying resource, expressed as a multiple of the speed of the standard platform. The job queue of a resource provider keeps an ordered set of jobs scheduled but not yet executed. Each job, once it is executed on a resource, will run in a dedicated mode on that resource, without time-sharing or preempting. A provider charges for a job according to its unit price and job length. Unit price refers to the price that the resource offers for executing a job of unit length. When a provider with capability 5 bids to execute a job of length 20 at a unit price of 2 and if the consumer accepts the bid and decides to send the job to run there, the job will take.

A multiprocessor system is defined with shared cache that is been accessed by multiple processor at the same time. There are number of associated terms to improve the cache access in terms of reliability and efficiency. As we know, the cache is been defined with small capacity, but in such system the frequency of user request is high so that there is the requirement of replacement of cache contents. This includes the incurring the latency because of memory access time.

II. REVIEW OF LITERATURE

Song Hao[1] has defined a prediction based approach for the L2 cache management and to optimize the page access. The author has defined the work for multi processor cache system with shared cache. Author defined the novel prediction approach to access the cache contents and reduce the memory blacks and closed blanks over the memory. Author also perform the analysis of proposed approach on HMTT toolkit and capture the memory trace to evaluate the cache latency. The obtained results shows that the pressed work has reduced

the latency upto 8.4%. The main improvement in the work is by performing the directional analysis on each processor in both horizontal and vertical direction. Another work on cache replacement is presented by Anupam Bhattacharjee[2] in year 2005. Author has defined an improvement over the LRU algorithm with the inclusion of randomized algorithm with random replacement of the page. The defined algorithm is based on the logical division of the cache and perform the distribution of error analysis on each division separately. The data structure adapted by the author is complex but it reduces the latency and optimized the cache access. Author implemented the concept for web page replacement. The work also having the assumption about the number of page banks in the pages. It also includes the utility function and reduces the fetching cost and the traffic latency. In year 2006, A. Mahjur[3] has defined the prefetching scheme to reduce the miss ratio for the caching. The author has defined a hardware based scheme in which complete memory is divided in smaller segments and perform the prefeteching over each block separately. This approach is based on the prediction mechanism in which a two phase algorithm is been defined based on markov model approach to reduce the page access latency and to reduce the miss ratio. This approach is effective for the local predictors and it help to differentiate and reduce the miss sequences. Kiyofumi Tanaka[4] has defined memory architecture based on gated control approach on each cache block with the concept of data compression. The presented work is about to reduce the energy consumption with secondary cache memory with memory leak. The author defined work to reduce the memory leaks on secondary cache and perform the successful implementation with upto 20% reduction in the leakage. Another work in year 2008, presented by Naizheng Bian[5]. In this paper author study the advantages and disadvantages of existing page replacement algorithms. The author has defined a new algorithm called Least Grage Replacement. This algorithm is based on the frequency and the history analysis. The author has implemented the work for web page replacement and analysis of the work is been defined under different parameters such as Hit Ratio, Byte Hit ratio etc.

In year 2008, Yingjie Zhao[6] has defined the page replacement scheme based on frequency and latency analysis. The author perform the analysis on existing hit ratio and based on it, defined the page replacement scheme so that miss ratio is been reduced. Author defined a sequential access to the system so that the miss penalty will be reduced and mean access latency is also reduced. In year 2010, Mesut Meterelliyoz[7] defined a cache memory based improved architecture. Author work on 6-T and 8-T SRAM that includes the features like stability and noise ineffective access to the cache. The author also defined the work under the temporal gradients. Author defined the analysis with the inclusion of micro sense base amplifier with DRAM cache with the robustness of read operation on cache memory. In year 2011, Stefano Di Carlo, 2011[8] defined a transform based approach for the page replacement by improving the LRU approach. Author work on a multi processor system and divide the available cache in smaller blocks. The author performed the cache analysis under the instruction and data caching separately. Author defined the improvement over the CMOS technologies by reducing the memory faults over the cache. In year 2011, Mohamed Wassim Jmal[9] defined a work based on elliptical curve based algorithm to improve the aching process. The presented work is divided in stages. In first stage, the work is defined for open source embedded software to generate the architecture. The architecture adopted here is the architecture cache that includes the cache configuration and adapt the embedded processor system with the inclusion of cache minimization. Author also perform the analysis respective to the cache size to analyze the performance degradation. The objective of this study is to integrate digital signature on a chip using an embedded processor while increasing the speed of treatment and minimizing costs. Virtual prototyping is an interesting designing method which allows the designer to validate the design without needing Hardware implantation. In year 2011, Korde P.S.[10] has defined a work on the management of cache memory. The presented approach defined a recussive approach on two type of cache and performed a matrix based analysis to reduce the miss ratio. The model adapted in this work include the assumption on cache access and to configure the optimal replacement policy. Author defined the future analysis based approach to perform the effective cache replacement. In year 2007, Sungjune Youn[11] defined architecture level work on multiprocessor based embedded system with the organization of the cache. L2 cache optimization includes the performance and reduce the memory consumption. L2 cache organization is based on the private cache and improve the concept of shared cache also. The on chip improvement is been defined to reduce the latency. The implementation of work is performed on CMP simulator. The work reduced the latency upto 27%.

III. RESEARCH METHODOLOGY

The effectiveness of the cache operation is based on a property of computer programs called locality of reference or locality principle. Analysis of programs shows that most of the program time is spent on executing many instructions repeatedly. These instructions may be a simple loop, nested loops, or a few procedures that repeatedly call each other. Many instructions in localized areas of the program are executed repeatedly during some period of time, and the remainder of the program is accessed relatively infrequently. This is referred to as locality of reference.

Locality of reference happens in two ways: temporal and spatial. Temporal locality means that a recently referenced memory word will be referenced again very soon. Spatial locality means that memory words close to recently referenced memory word will be referenced soon. Temporal locality happens when server executes iterative loops and calls to subroutines. Spatial locality is seen when the server performs operations on tables and arrays The memory circuitry is designed to take advantage of the locality of reference. The temporal locality suggests that whenever a word is first needed, it should be brought to cache where it will hopefully remain until it is needed again. The spatial locality suggests that instead of fetching just one word from the main memory to the cache, several words adjacent to the needed word are also fetched into the cache.

The proposed work is about to define a cache data replacement scheme based of frequency analysis. According to this scheme most frequent data items will be kept in cache. As the most required data items are in cache itself the system will improve the efficiency as well as will improve the hit ratio. The system is presenting an efficient and reliable data replacement scheme in cache memory. To work with the proposed system we have to simulate the service allotment in Multiprocessor market based architectural environment in a programming environment. Here the work is basically about the improvement over the process allotment. The presented work will show the access to the cpu with access time and the relative parameters. As the architecture build up the next work is to define any of the existing data replacement algorithm in cache memory. The work is here based on the access of data from internal cache to cpu. The analysis will be done on this existing approach. Now the proposed work will be implemented where we will a list to store the frequencies of the dataset. Now according to the user access the frequency of data items will be changed. And the data values in cache with low frequency will be replaced by the high frequency data items.

The presented scheduling mechanism is given as under

```
Step 1:   Initialize the Page Cache of Size N
Step 2:   Perform User Input Page called Pagei
Step 3:   if count(inputpage)<=N
{Include the Page in Queue}
Step 4:   else
{If Pagei   BelongsTo  Queue
{Frequency(Pagei)=Frequency(Pagei)+1}
Step 5    Else
{Find the Low Frequency Page From Cache called LFrequPage
If Count(LFrequPage)=1
{Replace this page by pagei}
5.3       Else
{Find the Recently Visited Page called RecVisited from LFrequPage and
replace it in cache by pagei}}}
Step 6    Exit}
```

IV. RESULTS

The presented model is implemented in matlab 7.8. The implementation is here been performed for the proposed algorithm. The analysis of the work is been performed for hit ratio and miss ratio analysis. The results obtained from the system are shown in figure 1.

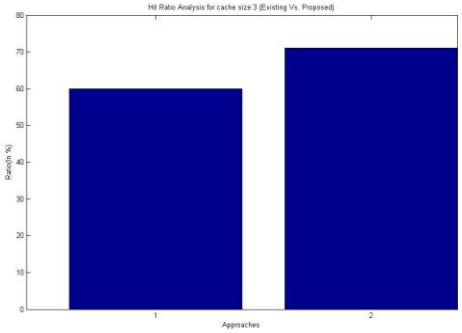

Figure 1 : Hit Ratio Analysis for Cache Size 3

As we can see, in figure 1, the hit ratio analysis of existing and proposed work is shown. In this proposed work a two level analysis approach is implemented. The cache size taken in this figure is 3. As we can see, in existing approach the hit ratio 60% whereas in proposed work hit ratio is 71%.

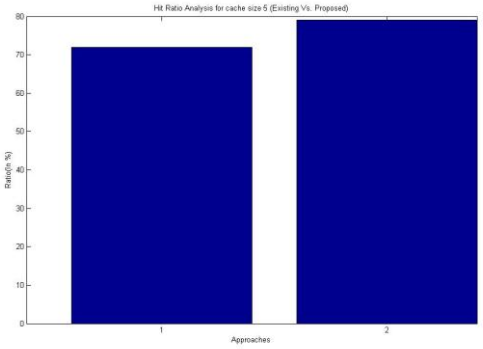

Figure 2 : Hit Ratio Analysis for Cache Size 5

As we can see, in figure 2,the results are obtained for cache size 5. Here the hit ratio in existing work is 72% but the proposed work gives the improved with higher hit ratio with 79%.

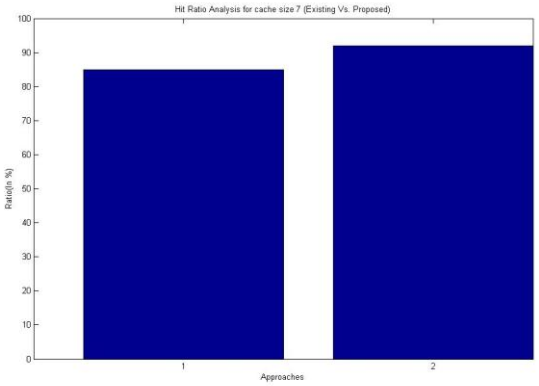

Figure 3 : Hit Ratio Analysis for Cache Size 7

As we can see, the results obtained from the system for cache size 7. The existing provides a lower hit ratio of 85% and the proposed work provides the hit ratio of 92%.

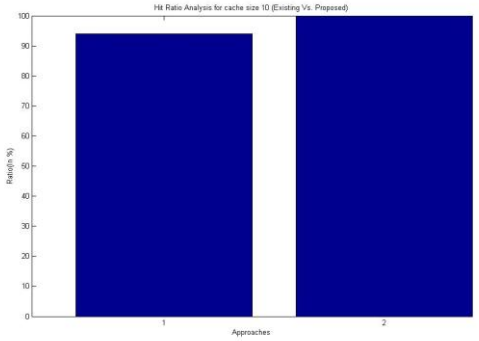

Figure 4 : Hit Ratio Analysis for Cache Size 10

In figure 4, the hit ratio analysis for cache size 10 is shown. The existing work gives the hit ratio 94% whereas proposed work gives the hitratio 100%.
From these results we conclude that the presented work is effective for different cache sizes. For each cache size the presented work is more effective than existing.

V. CONCLUSION

In this paper, an improved caching mechanism is been represented. The presented approach is two level approach in which at first the prioritization of cache is been performed. Later on the analysis of replaced page is been done using the request frequency and utility analysis. The results are presented in the form of graphs.

REFERENCES

[1] S. Hao, Z. Du1, D. Bader,M. Wang A Prediction based CMP Cache Migration Policy", *The 10th IEEE International Conference on High Performance Computing and Communications* , 374-381,2008.

[2] A. Bhattacharjee, B.K.Debnath,, A New Web Cache Replacement Algorithm, Communications, Computers and signal Processing, 420-423,2005.

[3] A. Mahjur, A.H. Jahangir,Two-phase prediction of L1 data cache misses, *IEEE Proc.-Comput. Digit. Tech, (153)(6*), 381-388, 2006.

[4] K. Tanaka, A. Matsuda, Static Energy Reduction in Cache Memories Using Data Compression, *In Proc. of TENCON*, 1-4, 2006.

[5] N. Bian,, H. Chen. A Least Grade Page Replacement Algorithm for Web Cache Optimization, Workshop on Knowledge Discovery and Data Mining, 469-472, 2008.

[6] Y. Zhao, N. Xiao, Saber: Sequential Access Based cachE Replacement to Reduce the Cache Miss Penalty, *The 9th International Conference for Young Computer Scientists*, 1389-1394, 2008.

[7] H.H. Harchegani, A. Farahi, H.M. Shirazi, A. Golabpour, P. Almasinejad, Using Genetic Algorithm for Prediction of Information for Cache Operation in Mobile Database, *Second International Workshop on Knowledge Discovery and Data Mining*, 148-151, 2009.

[8] M. Meterelliyoz, J.P. Kulkarni, K. Roy, Analysis of SRAM and eDRAM Cache Memories Under Spatial Temperature Variations, *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, 2-13, 2010.

[9] S. D. Carlo, P. Prinetto, A. Savino, Software-Based Self-Test of Set-Associative Cache Memories, *IEEE Transactions of Computers*, 1030-1044, , 2011.

[10] M.W. Jmal, W. Kaaniche, M. Abid, Memory Cache optimization for a digital signature program: case study, *8th International Multi-Conference on Systems, Signals & Devices*, 1-5, 2011.

[11] P.S. Korde, P.B. Khanale, Recursive Storage Cache Memory for Matrix Multiplication, Recent Advances in Intelligent Computational Systems, 581-586, 2011.