**Instructor**: Zeshan A. Chishti

**Student**: Achraf Dayoub

**Class**: ECE 585

**Date**: 02/22/2020

**Variation of Cache Replacement Policies for Different Applications**

1. Seon-Yeong Park1, Dawoon Jung, Jeong-Uk Kang, Jin-soo Kim, and Joonwon Lee, “CFLRU: A Replacement Algorithm for Flash Memory\*”, *CASES '06 Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems*, ACM New York, October 22, pp. 234-241, 2006.

This Paper presents its views on the implementation of Least Recently Used (LRU) policy in Flash Memory devices which finds its application in Embedded Systems. The author differs from the user of existing LRU policy used in Operating System in magnetic devices with a new scheme called as Clean First LRU which considers of replacement cost caused by considering dirty victims and the excessive time taken by write operations. The Author forays into improving the performance by using LRU for requesting page from secondary memory and the use of clean page for eviction. These techniques are implemented in a Linux 2.4.28 machine with two traces one from Val grind and the other from ext2 system of Linux. The results are compared with plain Linux, Linux kernel with Read Ahead policy and the flash read, write, time and the expected energy is calculated and plotted.

Reason for Selection: This paper provides a practical solution to the contemporary problem in the Embedded system. Also, this paper published in CASES conference has 481 citations in Google Scholar.

2. Qingbo Zhu, Asim Shankar and Yuanyuan Zhou, “PB-LRU: A Self-Tuning Power-Aware Storage Cache Replacement Algorithm for Conserving Disk Energy”, *ICS '04 Proceedings of the 18th annual international conference on Supercomputing,* ACM New York, June 26, pp. 79-88, 2004.

This paper provides a new cache replacement algorithm PB-LRU to save the power consumption in a Multi-Disk Storage Systems. The Authors try to overcome the cumbersome practices of the PA-LRU algorithm used earlier with many tunings. Here they try to implement PB (Partition Based) that requires little parameter tuning. The main idea they have achieved is to divide the storage cache into separate partitions one for each disk. With this information, they have devised a cache partitioning scheme by formalising it to Multiple Cache Knapsack Problem (MKCP) which is solved using dynamic programming. The Authors have used the DiskSim Simulator with two real and two synthetic traces and have evaluated the energy saving on each of the four traces by four different algorithms – Infinite cache, LRU, PR-LRU and PB-LRU. Finally, using the simulation results, they put forward their proof that PB-LRU provides better or similar outcomes in specific workloads on par with PA-LRU.

Reason for Selection: The Authors have provided a solution for an existing problem by streamlining an algorithm through Dynamic Programming, which is unique. Also, this paper has 106 citations in Google Scholar.

3. Akanksha Jain, Zhan Shi, Xiangru Huang, Calvin Lin, “Applying Deep Learning to the Cache Replacement Problem”, *MICRO '52 Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, ACM New York, October 12, pp 413-425, 2019.

This paper implements a Deep Learning methodology to optimise the cache replacement problem. The Authors have tried to assemble the multi-layer neural networks such as Recurrent Neural Networks which were ill-suited for hardware predictions to make a cache replacement policy. They have designed a Support Vector Machine (SVM) that represent a simpler linear learning model which is trained online in hardware and which matches the offline accuracy. With these insights, they have created a Glider Cache Replacement Policy. They have built this Policy by building on the Hawkeye Cache replacement policy, in which the cache-friendly data is inserted with high priority and cache averse data is inserted with low priority. Using this policy, an LSTM Model built to make optimal caching decision. This cache decisions are evaluated using SPEC CPU 2006 benchmarks and compared against online and offline models. The Accuracy of these models helped to devise Glider, the practical cache replacement policy that provides performance superior to the previous state of art policies.

Reason for Selection: The cross-platform approach of using Deep Learning to solve the existing hardware problem is path-breaking and opens new avenues for solving problems using different paradigms. Also, this recent paper provides a unique approach towards cache replacement instead of regular algorithmic optimizations.

4. Jaafar Alghazo, Adil Akaaboune, Nazeih Botros, “SF-LRU cache replacement algorithm”, *Records of the 2004 International Workshop on Memory Technology, Design and Testing,* IEEE, August 30, 2004.

This paper tries to implement a new technique called SF-LRU (Second Chance Frequency) using a combination of LRU (Least Recently Used) and LFU (Least Frequently Used). The algorithm not only counts the number of times a block is referenced like LRU but also give a second chance to the memory reference that have been referenced maximum. The authors have chosen to call this Recency Frequency Control Value. The authors have tried to compare this value when there is an eviction. They have also provided the performance evaluation which significantly amps the miss and hit rate of the LRU and LFU by a percentage of 6.3% and 9.3%. In miss and hit ratio. By reducing the mass ratio, the overall power consumption has been reduced.

Reason for Selection: The Authors have provided a novel method to combine the advantages of both LRU and LFU and use the corresponding gains to reduce the power consumption.

5. Ke Zhang, Zhensong Wang, Yong Chen, Huaiyu Zhu, Xian-He Sun, “PAC-PLRU: A Cache Replacement Policy to Salvage Discarded Predictions from Hardware Prefetchers”, *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing,* May 23 -26, 2011.

This paper puts forth a new PLRU policy named PAC (Prediction Aware Confidence) which uses the prediction result generated by the existing stride prefetcher to avoid the removal of the predicted cache in the near future. The reason they use this prefetcher is to prevent the eviction of the prefetched block when it is still available in the cache. The prefetched block is taken into the cache based on the prediction of which cache block will be accessed in the future. So, it is unnecessary to discard it from the cache and fetch it again. The Authors work further proposes the creation of confidence level of a predictor that calculates if a block will be 100% accessed in the future or if we are less confident about the use in the future and this is embedded as a hash function in the PLRU node. This proposed solution was simulated and tested using the SPEC CPU suite; the simulation output revel that PAC PLRU reduces the L2 miss rate by 91%. In addition to this, this method requires some additional hardware but those are relatively straight forward and insignificant in cost.

Reason for Selection: This paper published in the IEEE International symposium is effective by proposing a solution that is relevant and cost-effective before it does not involve a complete re-organization of the hardware of the existing hardware.

6. P. Jayarekha, 2T.R. Gopalakrishnan Nair, “An Adaptive Dynamic Replacement Approach for a Multicast based Popularity Aware Prefix Cache Memory System”, *arXiv arXiv:1001.4135v1, 2011.*

In this paper, the authors have explored the usage of dynamic replacement approach in Multimedia servers where each data can be a combination of video, text, audio and image data. These data can consume large space and they would consider high bandwidth when transmitted over a network. Therefore, the servers store a prefix of these videos in the cache memory which can be accessed swiftly. The authors have considered an age factor; this helps during eviction by considering both frequency of usage and the time it has been in the cache. In their approach, they have used two LRU page list which maintains if the page has been requested only once or twice and a ghost list which contains which keep track of recently evicted cache. They have simulated the values in the server and obtained the result which has a significant increase in the hit ratio, bandwidth and waiting time when compared to the Least Recently Used policy.

Reason for Selection: The authors have implemented the cache replacement policy for the problem of multimedia server by considering the relevance and the frequency of usage which gives a better throughput to the end-user.

7. S. Jin, A. Bestavros, “GreedyDual Web caching algorithm: exploiting the two sources of temporal locality in Web request streams”, *Elsevier, Computer Communications, v 24,* January 2001, pp 174-183, 2001.

In this paper, the authors have detailed how the web caches differ from the caches in the memory systems and provided the issues which are uniquely affecting the web caches. Also, the authors have put forth the different types of caching namely Network, Server and Client Cache which are available. Their work tries to overcome the disadvantage of the LRU policy which does not take into account different cache sizes. The Greedy Dual algorithm tries to overcome the issues with the previous methods by capturing the long-term popularity and rate of ageing. The algorithm is implemented using traces and compared against number of cache replacement policies. The outcomes of these Greedy Dual algorithm are calculated by two models namely constant cost and packet cost model. These models optimise the Hit rate and Byte hit rate. Based on the results obtained it can be concluded that the Greedy Dual remains competitive and significant outperforms the recency-based algorithm.

Reason for Selection: The paper differs in the approach of web caching by not utilising the same techniques used in-memory caching. And it has 259 citations in Google scholar.

8. Muhammad Bilal, Shin-Gak Kang, “Time-Aware Least Recent Used (TLRU) Cache Management Policy in ICN”, *16th International Conference on Advanced Communication Technology,* March 27, IEEE, 2014.

In this paper, the authors have proposed a Hybrid Time-based LRU replacement scheme and they have tried to use this scheme in Information Content Network (ICN) which are nothing but a network of caches. The Authors have made an analytical model in which when the consumer has requested a data or put forward a data into a node then the data gets cached into the network. The usage and the eviction policies considered are similar to the LRU, but the authors have introduced a new terminology called TTU (Time to Live). The TTU value will decrease and will become zero before the arrival of the next clock. The values are analyzed, and the results are compared for the states Ergodicity, Queuing Delay, Eviction time approximation. They have also performed the same operations under different processors, based on which they have concluded that TLRU is significantly better when compared to LRU by presenting the usage statistics in a graph.

Reason for selection: This paper published in the Conference on Advanced Communication technique provides ways to overcome the caching issues in-network caches. And is one of the unique documents which deal with time parameter in Cache Networks.