Project Proposal: Cache Replacement Policy

Ariana Pouya
828004580
Texas A&M University
College Station, Texas
ap25@tamu.edu

Michelle Chatter
720000594
Texas A&M University
College Station, Texas
mchatter@tamu.edu

Ashwin Kundeti
329009927
Texas A&M University
College Station, Texas
ashwin_k@tamu.edu

Sai Madhava
234008701
Texas A&M University
College Station, Texas
saimadhava@tamu.edu

***Abstract*—Cache replacement policies play a crucial role in optimizing cache performance by determining which data to evict when the cache reaches capacity. Traditional strategies like Least Recently Used (LRU) and Least Frequently Used (LFU) struggle with dynamic workloads, causing inefficient eviction decisions. This project explores an enhanced cache replacement strategy leveraging Signature-based Hit Predictor (SHiP) to improve eviction accuracy. SHiP leverages signature-based learning to adapt dynamically, prioritizing cache retention based on past access patterns. This approach aims to enhance cache hit rates and overall system efficiency. The proposed method will be evaluated through simulation-based analysis, comparing its performance against existing replacement policies to quantify its effectiveness.**

***Index Terms*—Cache, RRIP, LRU, LFU, SHiP**

I. Introduction

Cache memory plays a crucial role in modern computing systems by acting as a high-speed intermediary between the central processing unit (CPU) and main memory. It reduces data access latency by storing frequently accessed data closer to the processor. However, due to its limited size, efficient cache management is critical for maintaining system performance. Cache replacement policies are designed to determine which data should be evicted when the cache becomes full, ensuring that the most relevant data remain accessible while minimizing latency.

Traditional cache replacement algorithms, such as LRU and LFU, have been widely adopted due to their simplicity and effectiveness in specific workload scenarios. However, these methods often struggle with dynamically changing access patterns, leading to inefficient cache utilization. As a result, these conventional strategies can cause unnecessary evictions, increasing the overall cache miss rate and reducing system efficiency.

To address these challenges, researchers have explored predictive cache replacement policies that estimate the likelihood of future accesses based on past access patterns. One such approach is SHiP [2], which enhances cache efficiency by leveraging signature-based learning to track and predict data reuse patterns. SHiP assigns unique signatures to cache blocks based on program behavior, allowing the system to make informed eviction decisions. By adapting dynamically to different workloads, SHiP outperforms traditional static policies such as LRU, leading to better cache retention and reduced miss rates.

In this project, our aim is to replicate and analyze the SHiP model to evaluate its impact on cache efficiency. Our approach focuses on measuring cache miss reduction and overall performance improvements. Through comparative analysis and simulation-based evaluation, we will assess how SHiP performs against traditional cache replacement policies and validate its effectiveness in improving system performance.

II. Objective

The objective of this project is to replicate and analyze an improved cache replacement policy based on SHiP [2]. Efficient cache management is crucial in modern computing architectures. Traditional approaches such as LRU often struggle to adapt to diverse workload patterns, affecting their effectiveness in real-world scenarios. This study aims to implement and evaluate the proposed SHiP based strategy under simulated workloads. The effectiveness of SHiP will be assessed by comparing its performance with LRU and newer policies by analyzing key performance metrics such as cache hit rate and overall system efficiency. This project validates the original study's findings to reveal the practical applicability and impact of adaptive cache replacement strategies on computational efficiency, thus providing insight into cache replacement strategies and their role in enhancing computational efficiency. The results of this study can be valuable in environments that rely on large-scale data processing and high-performance computing, including operating systems, cloud infrastructure, embedded systems, and artificial intelligence workloads, where efficient cache management is critical for optimizing performance.

III. Background and Motivation

Modern processors rely on cache memories as a crucial component of the memory hierarchy. Caches bridge the speed gap between the CPU and main memory, impacting overall system performance. However, traditional replacement policies such as LRU and LFU have limitations with dynamic workloads. The decisions made are often not optimal and lead to more cache misses and higher latency.

Research has explored adaptive replacement policies to overcome these limitations. For example, the Expected Hit Count policy [3] establishes a correlation between a cache block's reuse distance and expected future accesses. By leveraging historical access information, EHC enhances victim selection while keeping hardware overhead minimal. With

workloads becoming increasingly heterogeneous, dynamic and adaptive cache policies are gaining more attention. In the instruction-controlled replacement policy for RISC-V processors [4], cache behavior is managed at runtime through specialized instructions, allowing software-driven fine-tuning of cache replacement strategies for different workloads.

SHiP [2] introduces an innovative approach to cache management by utilizing program behavior signatures rather than relying solely on recency-based heuristics. Unlike LRU, which assumes recent accesses are the best predictors of future reuse, SHiP tracks cache block reuse patterns using instruction signatures. This method allows fine-grained predictions, ensuring that frequently reused blocks remain in the cache while identifying blocks that are less likely to be accessed again. SHiP significantly reduces cache miss rates while maintaining low hardware overhead.

Other proposals, such as the Re-Reference Interval Prediction (RRIP) and its variant RT-RRIP [1], introduce recency-based filtering to refine replacement decisions. RT-RRIP builds upon RRIP's foundational principles by using recency time prefilters to estimate the likelihood of future cache reuse.

Collectively, these advancements underscore the importance of predictable and adaptable cache replacement policies. While RT-RRIP refines recency-based prediction, SHiP moves towards a signature-driven approach, leveraging historical access signatures to provide more precise predictions with minimal hardware cost. These innovations contribute to optimizing cache hit rates, reducing miss latency, and adapting to diverse workload behaviors, ultimately enhancing overall system performance.

IV. Relevance

Cache replacement policies play a crucial role in processor efficiency by reducing cache misses and improving memory performance. Efficient cache replacement policies are critical for modern computing, as poor eviction decisions lead to higher miss rates, inefficient cache utilization, and increased system latency. Traditional policies like LRU and LFU struggle with scan-based and dynamic workloads, often failing to adapt to unpredictable access patterns. This causes cache thrashing, reducing system efficiency. SHiP addresses these challenges by introducing a signature-based learning approach, which allows it to track instruction-specific access patterns and predict reuse more accurately. SHiP dynamically adapts to changing workloads, ensuring better cache retention and reducing unnecessary evictions. Validating SHiP's advantages will enhance cache management for modern processors, optimizing memory access for higher efficiency, reduced miss latency, and lower power consumption. By incorporating signature-based learning, SHiP provides a more adaptable and predictive cache replacement strategy, making it a valuable improvement over conventional policies.

V. Methodology

The objective of this project is to evaluate the performance of the SHiP cache replacement policy in an three-level memory hierarchy. The tasks required to execute this project are broken down into five phases. Each member of the team will be responsible for developing the algorithm for one replacement policy and then integrating it into the architecture. Each team member will be responsible for simulating and evaluating the performance of all replacement policies with a set of SPEC benchmarks. Consolidating the final results and developing the final documentation will be a collaborative effort. Specific responsibilities will be adjusted throughout the project as needed.

1) Phase 1: Develop a three-level cache hierarchy system model that resembles the Intel Core i7 system.

2) Phase 2 : Recreate the SHiP algorithm as well as the other cache replacement policies used by [2] for comparison, including LRU, Seg-LRU and RRIP.

3) Phase 3: Set up the hardware simulator, such as ChampSim or ZSim, to evaluate the performance of the architecture with the different replacement policies using a subset of "Standard Performance Evaluation Corporation" (SPEC) benchmarks selected by [2].

4) Phase 4: Evaluate the performance of each replacement policy used in the different benchmarks by calculating the IPC, miss rate, and speedup.

5) Phase 5: Document and report the process and findings of the experiment.

VI. Expected Outcome

Based on the findings, the expected outcome of this project is that SHiP based cache replacement strategy will demonstrate higher cache hit rates and lower miss rates compared to the other replacement policies [2]. Through simulation-based evaluation, we anticipate that the adaptive nature of SHiP will enable more efficient cache utilization, resulting in a lower frequency in cache misses and a reduction in miss latency. The application of this replacement policy will attribute to improved memory access performance and overall system efficiency in the benchmarks.

References

[1] C. R. Athni, V. V. Chippalkatti, A. Nandakumar, N. A. V, and P. Y. J, "Improved Cache Replacement Policy Based on Recency Time Re-Reference Interval Prediction," in *Proceedings of the 2022 IEEE 7th International Conference for Convergence in Technology (I2CT)*, Pune, India, Apr. 2022.

[2] C.-J. Wu, A. Jaleel, W. Hasenplaugh, M. Martonosi, S. C. Steely Jr., and J. Emer, "SHiP: Signature-based Hit Predictor for High Performance Caching," in Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2011.

[3] A. Vakil-Ghahani, S. Mahdizadeh-Shahri, M.-R. Lotfi-Namin, M. Bakhshalipour, P. Lotfi-Kamran, and H. Sarbazi-Azad, "Cache Replacement Policy Based on Expected Hit Count," in *IEEE Computer Architecture Letters*, vol. 17, no. 1, 1 Jan.-June 2018, doi:10.1109/LCA.2017.2762660.

[4] R. Takayama and J. Tada, "An Implementation of an Instruction Controlled Cache Replacement Policy on a RISC-V Processor," in *2023 Eleventh International Symposium on Computing and Networking Workshops (CANDARW)*, Matsue, Japan, 2023, pp. 172–178, doi: 10.1109/CANDARW60564.2023.00036.