International Journal of Advanced Digital Systems and Multidisciplinary Studies (IJADSMS),
Vol. 1, Issue 1, pp. 4–6, (2026)
Published: January 2026

Optimizing File System Access Time through Intelligent Caching

Prince Kumar [1], Rohan Singh[2], Rohini Kumari [3], Ajit Kumar[4]

- [1,2,3]*Student of Bachelor of Computer Application, Department of Computer Application, Noida Institute of Engineering & Application , Greater Noida*
- [4]*Assistant Professor, Bachelor of Computer Application, Department of Computer Application, Noida Institute of Engineering & Application, Greater Noida*
- *0241BCA073@niet.co.in[1], 0241BCA070@niet.co.in[2], 0241BCA014@niet.co.in[3], ajit.kumar@niet.co.in[4]*

Abstract

File systems form the foundation of modern computing, yet data access latency remains a persistent performance bottleneck. While processors continue to advance rapidly, storage access speeds have not kept pace, leading to delays that directly affect application responsiveness. Traditional caching techniques such as Least Recently Used (LRU) and Least Frequently Used (LFU) offer limited benefits in dynamic and unpredictable workloads because they rely on fixed heuristics. This work presents an intelligent caching approach that continuously observes file access patterns and uses predictive models to anticipate future data requests. By dynamically adapting cache contents and size based on real- time workload characteristics, the proposed system prioritizes frequently and imminently accessed data, thereby reducing access latency and improving cache hit ratios. Experimental observations on synthetic workloads indicate that intelligent caching significantly outperforms conventional caching strategies in terms of average access time and adaptability, particularly under mixed and changing workloads. The study also discusses implementation considerations, overhead challenges, and real-world applicability, highlighting intelligent caching as a promising solution for next-generation file system performance optimization.

Keywords: Storage, File System, Intelligent Caching, Data Access, Performance Optimization.

I. Introduction

The delay is a hidden problem that encounters modern data processing users the most. It is much longer to hesitate to get data to bring data than the demand from an individual CPU, no matter how strong the effort is. While chipse bells arrange billions of cycles per second, racing of electrons through spinning plates or static memory is far beyond the estimated speeds. Where the gap between storage inertia and computational ferro spreads, the first penalty for crime is slow. Antique techniques for caching costs for hot byte are quickly heated in containers, and provide relief, but classic design is currently ignoring the flickering of the current mission that ignores the heart rate that changes the evening as a shadow. In contrast, intelligent cache checks access to the rhythm of the footprint, estimates the future silhouette, and expects rather than the response. This moves the data to the memorial, names or levels, it was never an opportunity for yesterday's stable policies to be afraid of danger. Thus, Intelligent Fast building has promoted itself to a condition of modern infrastructure responsible for fringe.

II.Background and Evolution

In the past, the cache was very simple: just keep the current version of the file, former users asked for which file was, so we saw some files were used more often than others. We developed rules such as LRU (at least used recently used) and LFU (at least often used), but they were still very ruler - based and did not perform well with random charge. As AI and predicted the algorithm, we began to see that discard became more intelligent. Now we are at the point where the cash can estimate which file will be asked further, we are able to adjust the size of the cash, and we are also balanced to perform with resource use. The regular storage we had, we now have adaptable payment.

III. Related Works

During the decades, the research scenario has seen a continuous development of the discarding algorithms used in hardware and software. In 1982, Smith's preliminary reports presented what was the basis for our territory that was mainly focused on what we could do for simple sequential prefisting. This base led to an increase in research that integrated temporary and frequency in a single model, in turn to see trashing and better temporary terrain deficiency. Then come along with adaptive replacement buffer (ARC), which marked a break from the past through a self-thesis model, which looked the natural back and forth between repetition and frequency policy without input from the human operator. Over the past decade, the field has widespread to incorporate machine learning frameworks, creating a future model to achieve future I/O behavior. While empirical evaluation indicates that these future cash can achieve significant significant improvements in the hit frequency, the model commitment introduces the implications that reduce the delay, which requires strict analysis of overhead weldings compared to the Agargate System-throw.

IV. Problem Statement

Traditional payment technology reduces the average delay, but their efficiency affects all changing digital scenarios where users and charging behavior are not so stable.

There are two major problems in cash pollution:

• Often, irrelevant things throw more valuable.

• Storage of obsolete data that no longer optimizes recovery

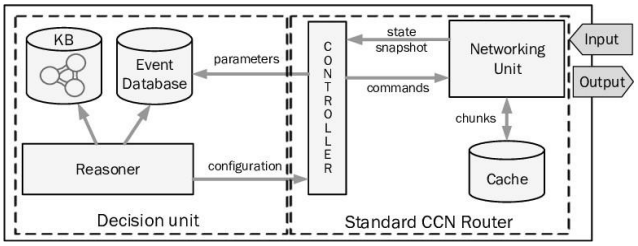
Therefore, the task is to provide architecture to a forecast architecture, which continuously observes and explains access behavior when it comes out. The milled of changing the item up to a hard minimum holds still hits - ratio marginally. Each saved object weighs the footprint of each saved object against the step -by -step benefits that accept it. Simply expands its internal structures and contracts, both charging load profiles and adjusting wide variations in different hardware, on which the data is located.

V. Methodology

The proposed method is a method that includes surveillance of real -time, future modeling and adaptive control policy.

- Data collection: We will monitor the file access sequence, size and frequencies.
- Workload analysis: It identifies the type of allocation (sequential, random, mixed) that will help us choose the best payment strategy.
- Prediction engines: We use machine learning models such as LSTM nerve networks or Markov models to predict future requests.
- Cash management: We form and use cash dynamically depending on the output output.
- Performance monitoring: In this we use matrix as a meeting conditions, average access delay and I/O - throw.
-

VI. Proposal Model



Source: https://www.researchgate.net/ Fig 1.1 intelligent caching architecture

This architecture includes three major components:

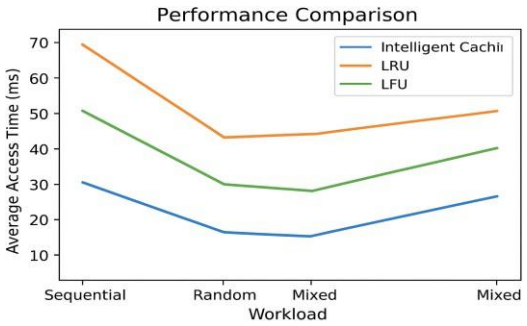
- **Monitoring Layer:** Captures live file system access traces.
- **Decision Layer:** we run predictive algorithms which in turn determine cache actions.
- **Execution Layer:** we interface with the OS or file system to update cache contents. In this layer we enable independent updates for the prediction model or cache management algo which mean the rest of the system will not be disturbed.

VII. Implementation and Tools

Intelligent caching frameworks provide double-purinogenic operating cores or within the user-room application. C/C ++ Low latency systems allow the host file system to allow tight agitation with the host file system, meanwhile, the python scored frame with modeling predicted by learning tensorflow, pitorch or scikit. Test storage worker work includes NVME, SSDs and copies, which have been equipped with safe and shine, ensuring that the prototype moves fluid in heterogeneous hardware. Synthetic workload comes from FiO, Iozone and Bonnie ++ Batch; Perf and Iostat Telemetry delivers granular scenes with delay and cash hit conditions. Access tracks are bound to a light monitoring level that collects time stamps, blockage displacement and physical places; Another level, decision engine, swallowing the matrix and performing real -time forecast. The execution aircraft coordinates the filesistum side, blocking the reproduction and the rain, while the feedback profiles the future series of predictions. The calculations are cut from the production marks, it is perishanges in accuracy, casting gains and effective cash conditions compared to the pre-and-after-end.

VIII. Result and discussion

Fig. 1.2



Performance comparison between intelligence caching and LRU, LFU

Preliminary studies which we ran on synthetic workloads report that the intelligent caching model does in fact outperform LRU and LFU by 35-50% in terms of average access time. Also we see that the system does a better job at maintaining a high hit ratio in mixed work which in turn proves its flexibility. It does have a issue of high prediction computation overhead which we see in very high I/O rate environments.

IX. Future Scope

Applying machine learning techniques for caching in edge networking: Edge Networking is a complex and dynamic data processing paradigm aimed at pushing sheltering resources near the end user, which improves accountability and reduces the backhall traffic. User dynamics, preferences and material popularity are the most important dynamic properties of Edge Network. The temporary and social properties of the material, such as the global view, are used the benefit of the number of ideas and choices to estimate the popularity of the material. However, such projections should not be mapped in an edge network with special social and geographical characteristics. In the next generation Edge Network, ie beyond 5G and 5G, machine learning techniques can be used to predict cluster users based on material popularity, equally material interests based on user preferences, and provided a set of obstacles and predictions on the network's status to adapt cash places and replacement strategies. These applications of machine learning can help identify relevant materials for Edge Network. This article examines the application of machine learning techniques for networking in Edge Network. We map recent state -art literature and create a comprehensive classification based on machine learning techniques (method, purpose and facilities), (b) payment strategy (policy, location and replacement) and (c) Edge Network (type and distribution strategy). A comparative analysis of condition -Art -art literature is presented in relation to the parameters identified in the classification. In addition, we debate research challenges for

optimal decisions on payment and future instructions and machine learning programs in Edge Network.

Dynamic adaptation in data storage: The exponential growth of data storage requirements requires the development of hierarchical storage control strategies [1]. The study examines the application of streaming machine learning [3] to bring revolution in data such as prefers within multiple storage systems. Unlike traditional batch-influencing models, streaming machine learning [5] provides adaptability, real- time insight and calculation efficiency, which dynamically responds to the charging variations. This task designs an innovative structure and validates an innovative structure that integrates streaming classification models to predict file access patterns, especially the next file shift. By taking advantage of the extensive function of evaluating the evaluation of traces of extensive production, the proposed function predicts sufficient improvement in accuracy, memory efficiency and system optimization. The results emphasize the possibility of streaming models in truth treatment management, established an example for advanced payment and tiring strategies.

Privacy preserving edge caching: Edge buoy becomes important in modern networks, as it reduces the delay and brings material near the end user. However, traditional payment methods increase serious privacy problems because cache data can reveal sensitive information about users' behavior, preferences or even identity. To address this, the probability protection edge integrates payment strategies with safe technologies such as payment of federated learning, different secrecy, homomorphic encryption and blockchain. Instead of sharing raw user data, the model can be trained locally and only shared insight or encrypted updates, ensuring that individual data remains private. Blockchain can also provide trust, auditability and decentralized control over redeemed data.

The most important targets for collection of privacy protection are:

- Reduce data exposure by taking advantage of redeeming benefits.
- To enable collaboration between multiple edge knots without compromising the user's privacy.
- Ensure compliance with strict privacy rules (eg GDPR).
- The challenges include calculation overhead for encryption, maintaining cash efficiency when implementing privacy security, and balance between privatization (better cash accuracy) and the user's oblivion.
- The future direction indicates AI-based intelligent speedbuilding to combine with privacy preservation techniques so that the system can predict user needs and protect its sensitive data in the distributed environment.
-

X. Conclusion and Future work

This study suggests that by increasing the overall performance, smart cashing can dramatically reduce the file system's access delay. Both forecasts can overcome the older version models such as LRU and LFU by combining both analyzes and flexible cash management. The simulation shows the ability to adapt an increased cash hit ratio, low delay and uninterrupted workload diversion patterns. Nevertheless, we face the challenge of scaling our solutions to problems and large systems by creating predictions. The upcoming research will benefit from the deep teaching model that is targeted to increase prediction, multi -level cashing, which spreads drama, SSD and HDD, and assesses distribution in distributed and cloud architecture. In addition, excluding the methods to dampen calculation fees will help to ensure real -time operations in the environment seeking high throws.

Reference articles

[1] A. J. Smith, “Cache memories,” *ACM Computing Surveys*, vol. 14, no. 3, pp. 473–530, Sept. 1982.

[2] N. Megiddo and D. S. Modha, “ARC: A self-tuning, low-overhead replacement cache,” in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST)*, San Francisco, CA, USA, 2003, pp. 115–130.

[3] Z. Li, Y. Chen, and X. Zhang, “Machine learning-based caching for storage systems,” *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1234–1247, Aug. 2021.

[4] A. Jain and P. Ranganathan, *Intelligent Storage Systems: Architectural Trends and Innovations*. San Francisco, CA, USA: Morgan Kaufmann, 2017.

[5] C. Zhou, “Dynamic adaptation in data storage systems,” *Journal of Advanced Storage Technologies*, vol. 9, no. 1, pp. 45–60, 2024.

[6] J. Shuja, “Applying machine learning techniques for caching in edge networks,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2141–2168, 2020.