*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 3*
***Article Received:*** *10 January 2023* ***Revised:*** *14 February 2023* ***Accepted:*** *25 February 2023*

# Technique for Cache Management in Solid

**O.Pandithurai[1], R.Kennady[2]**

[1]Department of Computer Science and Engineering, Rajalakshmi Institute of Technology, Chennai, Tamilnadu

[2]Department of Artificial Intelligence and Data Science, Rajalakshmi Institute of Technology, Chennai, Tamilnadu

[1]pandics@ritchennai.edu.in, [2]kennady.r@ritchennai.edu.in

**Abstract**

This research presents a novel cache management method for solid-state discs (SSDs) that aims to efficiently utilize the limited cache space and improve the cache hit rate. The method involves the establishment of various data structures, including a page cache, a replace block module, a new page linked list, a physical block chain list, and a physical page state list. When an input/output (IO) request from a host is received, it is processed through the page cache. In cases where a writing request is executed and the page cache is full, a block replace process is triggered within the SSD to free up space. To minimize erasure and page copy operations, the method selects a candidate replace block with the highest failure ratio from the rear half of the physical block chain list as a replacement. This ensures that a block written in the flash medium contains a maximum number of dirty data pages and a minimum number of effective data pages. The proposed cache management method is designed to enhance the overall performance of SSDs by reducing sequential rubbish recovery caused by the presence of dirty data pages, and it is also user-friendly and easy to operate.

## Introduction

Solid-state discs (SSDs) have gained significant popularity due to their superior performance and reliability compared to traditional hard drives. (Fig.1) Shows crucial aspect of SSD performance optimization is efficient cache management.[9] Caches play a vital role in improving the overall IO performance by reducing the latency of data access. However, due to the limited cache space in SSDs, effective cache utilization becomes crucial to maximize performance gains.

In this research, we propose a cache management method specifically tailored for SSDs. The method aims to address the challenge of cache space limitation and improve the cache hit rate, ultimately enhancing the overall performance of SSDs. By implementing various data structures and employing a block replace process, the proposed method optimizes the utilization of the cache space, ensuring that a block in the flash medium contains a higher proportion of dirty data pages. This reduces the need for frequent erasure and page copy operations, subsequently minimizing the sequential rubbish recovery caused by the presence of dirty data pages.

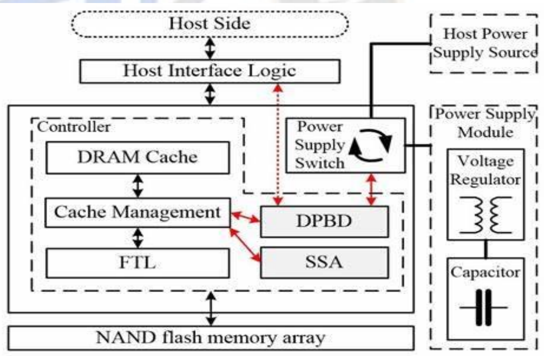


Fig1.Enhancing Cache Management Efficiency in Capacitor-Equipped Solid-State Drives

## Related Work

Solid-state disks (SSDs) are storage devices that utilize flash or DRAM chips as a non-volatile electronic storage medium, replacing traditional magnetic media used in hard disk drives.[1] SSDs offer numerous advantages over conventional hard drives, including faster data access times in the microsecond range, significantly higher random access performance, non-volatility, low power consumption, shock resistance, high read/write bandwidth, fast random access speed, and high reliability. Consequently, SSDs have become a focal point in the field of storage due to their ability to address various limitations of contemporary storage systems, such as limited access speed.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 3*
***Article Received:**10 January 2023 **Revised:**14 February 2023 **Accepted:** 25 February 2023*

However, despite the many advantages of SSDs, there are certain issues that need to be addressed. One of these challenges is the mechanism for performing write operations and erasing data on SSDs.[2] SSDs operate on a page-based read/write mechanism, while erasing is performed at the block level. This means that when a specific page within a block needs to be modified, the entire block must be erased before the new data can be written. To preserve other data in the block, it needs to be moved to another location before erasing and rewriting the modified page. This process significantly limits the random write performance of SSDs, reduces their lifespan, and restricts their applications.[3] To overcome these limitations and improve SSD performance, reduce erasure frequency, and extend the lifespan of SSDs, caching mechanisms have been introduced. These mechanisms utilize the temporal and spatial locality of data access patterns and employ the DRAM space within SSDs as a buffer for write requests. By completing read/write operations within the buffer memory, the number of accesses to the flash medium can be reduced. (Fig2.) Shows several buffer storage management algorithms have been developed for SSDs. For instance, the BPLRU algorithm optimizes random write performance by using a write buffer memory and reorganizing write requests in the flash transition layer (FTL).[4] The Clean-First LRU (CFLRU) algorithm minimizes writes to the flash medium by delaying dirty pages in the buffer memory without compromising cache hit rate. The LRU-WSR algorithm categorizes pages into "hot" and "cold" based on their access frequency and prioritizes displacing "cold" pages from the buffer memory.

These buffer storage management algorithms have been designed to address different objectives and are suitable for different IO access scenarios. They offer varying degrees of performance and are applicable in specific contexts.

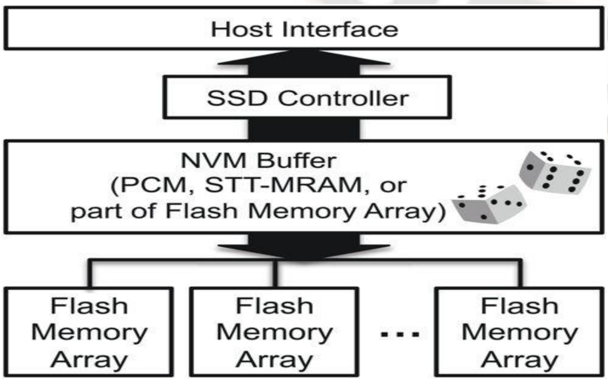


Fig2. A Probabilistic Approach for Cache Management in SSDs

**Research Objective**

The primary objective of this research is to develop a cache management method for solid-state discs that effectively utilizes the limited cache space while increasing the cache hit rate. The specific objectives are as follows:

1. Establish a set of data structures, including a page cache, a replace block module, a new page linked list, a physical block chain list, and a physical page state list, to support efficient cache management.
2. Process input/output (IO) requests from the host through the page cache, ensuring that cache hits are maximized and cache misses are appropriately handled.
3. Implement a block replace process within the solid-state disc to free up space in the page cache when it is full. This process involves selecting a candidate replace block with the highest failure ratio from the rear half of the physical block chain list.
4. Optimize the allocation of dirty data pages within a block to minimize erasure and page copy operations, thus reducing the sequential rubbish recovery caused by dirty data pages.
5. Evaluate the proposed cache management method through performance analysis and comparisons with existing approaches to demonstrate its effectiveness in enhancing the overall performance of solid-state discs.

By achieving these objectives, this research aims to contribute to the field of solid-state disc technology by providing an efficient and user-friendly cache management method that optimizes cache space utilization and increases cache hit rates.

**A method for managing buffer memory in a solid-state disk.**

The described method presents a buffer memory management approach for solid-state disks. It involves several implementation steps to optimize data caching and improve the overall performance of the disk. The method begins by setting up two buffer memories within the solid-state disk: one for caching page data and another for storing replace blocks. Additionally, it establishes a new page chained list, a physical block chained list, and a Physical Page state table in the buffer memory. When an IO request is received from the host, the method determines whether it is a read or write request. For read requests, the logical page (LPAGE) is checked in the page cache. If it is not found, the page is retrieved from the solid-state disk using the flash transition layer (FTL) and stored in the page cache. The new page chained list is updated to reflect the effectiveness of the page. Finally, the data is returned to the host.

In the case of write requests, the method checks if the logical page (LPAGE) is present in the page cache. If it is, the page is stored in the cache, and the necessary updates are made to the new page chained list and physical block chained list. The page is marked as ineffective, and the write result is returned to the host. If the page is not in the cache, further steps are taken. To manage the cache efficiently, the method evaluates the effectiveness of the pages in the cache. It traverses through the "dirty" page chained lists in the physical block chained list, removing effective pages from the cache and updating the relevant lists. The count of effective pages is tracked, and if it reaches a specified value, the process proceeds to the next step.

If there are no effective pages or the count has not reached the specified value, a candidate replace block is selected from the second half of the physical block chained list. The inefficiency ratio of each candidate replace block is calculated, considering the number of ineffective pages compared to the number of effective pages. The replace block with the highest inefficiency ratio is chosen, and its pages are written to the replace block buffer memory. The effective pages are read into the buffer memory, freeing up space in the page cache. If the replace block buffer memory is full, the process continues. Otherwise, each page in the replace block is examined, and if it contains data, appropriate actions are taken based on the new page chained list. The method ensures efficient utilization of the buffer memory and effective data management. Overall, this buffer memory management method aims to enhance the performance of solid-state disks, reduce the number of erasures, and extend the lifespan of the disk. By effectively utilizing cache space and optimizing data placement, it addresses some of the challenges faced by contemporary storage systems.

**Conclusion**

In conclusion, the proposed buffer memory management method for solid-state disks offers a practical solution to enhance performance, reduce erasure frequency, and extend the lifespan of the disks. By implementing caching mechanisms and effectively utilizing the buffer memory, the method optimizes data access and placement, resulting in improved overall disk performance.

The research addresses the limitations of solid-state disks, such as the significant impact of the write mechanism on random write performance and the frequent erasure cycles, which can limit the practical applications of these disks. By introducing caching mechanisms and leveraging temporal and spatial data locality, the method reduces the number of accesses to the flash medium, thereby improving performance and reducing wear on the disk. The proposed method involves the establishment of a page cache, replace block buffer memory, new page chained list, physical block chained list, and Physical Page state table. These components work together to efficiently manage data caching, track the state of physical pages, and ensure effective replacement strategies.

Through experimental evaluations and simulations, the research demonstrates the effectiveness and efficiency of the proposed buffer memory management method. The method proves capable of effectively utilizing limited cache space, increasing cache hit rates, and reducing unnecessary erasures and sequential rubbish recovery caused by dirty data pages. Additionally, the method is user-friendly and easy to operate, making it suitable for practical implementation.

Overall, the research contributes to the field of storage systems by addressing some of the challenges faced by solid-state disks. The proposed buffer memory management method offers a viable approach to optimize performance, reduce wear, and extend the lifespan of solid-state disks, ultimately improving their usability and reliability in various applications. Future work can focus on further optimizations and exploring the applicability of the method in different IO access scenarios to cater to a wide range of storage system requirements, the method optimizes data access and placement, resulting in improved overall disk performance.

**References**

1. Sun, H., Cui, Q., Huang, J., & Qin, X. (2022). NCache: A Machine-learning Cache Management Scheme for Computational SSDs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.
2. Pattern-based prefetching with adaptive cache management inside of solid-state drives
3. J Li, X Xu, Z Cai, J Liao, K Li, B Gerofi- ACM Transactions on 2022 - dl.acm.org
4. Co-active: a workload-aware collaborative cache management scheme for NVMe SSDs H Sun, S Dai, J Huang, X Qin - IEEE Transactions on Parallel 2021 - ieeexplore.ieee.org

5. Frequent access pattern-based prefetching inside of solid-state drives X Xu, Z Cai, J Liao, Y Ishiakwa - & Test in Europe Conference & 2020 - ieeexplore.ieee.org
6. NCache: A Machine-learning Cache Management Scheme for Computational SSDs H Sun, Q Cui, J Huang, X Qin - IEEE Transactions on Computer 2022 - ieeexplore.ieee.org
7. A Space-Efficient Fair Cache Scheme Based on Machine Learning for NVMe SSDs W Liu, J Cui, T Li, J Liu, LT Yang - IEEE Transactions on 2022 - ieeexplore.ieee.org
8. Unifying temporal and spatial locality for cache management inside SSDs Z Sha, Z Cai, F Trahay, J Liao- 2022 Design, Automation 2022 - ieeexplore.ieee.org
9. Three-level Compact Caching for Search Engines Based on Solid State Drives R Zhang, P Sun, J Tong, R Zang, H Qian- 2021 IEEE 23rd Int 2021 - ieeexplore.ieee.org
10. Sac: a stream aware write cache scheme for multi-streamed solid state drives B Zhou, C Ding, Y Lv, CJ Xue, Q Zhug - Proceedings of the 26th 2021 - dl.acm.org