Enhanced Database Caching Management through Virtual Memory: Innovative Research on Optimization Strategies of vmcache

Group 7

鄧雅文
R12921059
電機所碩一

王偉力
R12922116
資工所碩一

周哲瑋
B09502132
機械系大四

吳吉加
R12921093
電機所碩一

ABSTRACT

With the rapid development of Artificial Intelligence technology and big data analytics, the demand for high-performance database systems is increasing. Especially in today's scenario where AI and machine learning models require real-time analysis of massive datasets, traditional database management systems face significant performance bottlenecks when dealing with large-scale external storage data. This study focuses on improving the read and write efficiency of the vmcache database caching system, particularly optimizing in the two key areas of spin lock mechanism and lack of idle space management. By introducing mutex-based synchronization mechanisms, we expect to reduce unnecessary CPU consumption in multi-threaded environments. Additionally, we will design an effective idle space management strategy to reduce memory fragmentation and improve access efficiency. This research aims to enhance the access efficiency of database systems, providing a solid foundation for handling AI applications with large datasets.

PROBLEM DEFINITION

The core problem of this study is to conduct in-depth research and optimization on the synchronization mechanism and memory allocation strategy in the vmcache designed by the original authors. Because the original authors did not fully implement the code or handled it in a relatively simple way, this may lead to CPU resource waste, memory fragmentation, and low storage efficiency, especially in dynamically changing data loads. Our goal is to systematically analyze the existing architecture, identify key factors affecting read and write performance as well as transaction throughput, and implement targeted optimization measures.

MOTIVATION

In today's data-driven era, the applications of Artificial Intelligence (AI) and big data are rapidly evolving, creating an unprecedented demand for database systems capable of quickly accessing and processing large amounts of data. Especially in today's scenario where AI and machine learning models require real-time analysis of massive datasets, traditional database management systems face significant performance bottlenecks when dealing with large-scale external storage data. Additionally, with the recent rise in DRAM prices and the enormous amount of data, storing all data in memory would incur staggering costs. Therefore, the optimal solution would be to selectively use secondary storage devices such as HDD or SSD to store all data and write the currently needed data to memory for access. In this context, seeking to improve data access speed through software-level optimization without increasing hardware investment significantly becomes a crucial path to enhancing overall AI application performance. This study aims to explore how to address this challenge by improving existing database cache management strategies. Our goal is to optimize the data access process, enhancing both read and write performance as well as transaction throughput. By implementing new locking mechanisms improvements and intelligent virtual memory utilization, we expect to enhance the access efficiency of the database system, thus providing support for AI and other data-intensive applications.

BACKGROUND INFORMATION

The vmcache and exmap designed by the original authors demonstrate excellent performance, especially on fast storage devices. Through hardware-accelerated page translation and flexible locking mechanisms, the speed and efficiency of operations are significantly improved. However, there is still room for optimization in the implementation details.

MAIN APPROACH

1. Improve the lock handling mechanism: The current design of vmcache may trigger lock contention under high-

concurrency conditions, especially in multi-core processor environments. To reduce the performance overhead caused by spin locks, we propose using the wait/notify mechanism of C++ to handle lock contention issues. Such a change will allow the system to release CPU resources while waiting for the lock to be released, thereby reducing invalid CPU cycles and improving the overall efficiency and responsiveness of the system.

2. Adjust lock granularity: For data that is frequently read and rarely modified, lightweight read-write locks are adopted to reduce lock overhead.
3. Implement free space management: Effective storage space management is crucial for improving data access efficiency in database systems. The current vmcache design does not involve a free space management strategy. We propose introducing a free space management system that will index and track storage space, thereby improving storage space utilization and reducing data fragmentation.

EXPECTED OUTCOME

It is expected that through different optimization strategies, the processing speed of database transactions and read-write performance will be improved, and the results will be analyzed and discussed to explore the optimization effects and the reasons behind them.

REFERENCE

[1] Virtual-memory assisted buffer management. (n.d.). https://www.cs.cit.tum.de/fileadmin/w00cfj/dis/_my_direct_uploads/vmcache.pdf

[2] Viktorleis. (n.d.). Viktorleis/vmcache. GitHub. https://github.com/viktorleis/vmcache/tree/master?tab=readme-ov-file

[3] 精读论文：virtual-memory assisted buffer management. 知乎专栏. (n.d.). https://zhuanlan.zhihu.com/p/611616867