

소형 모바일 기기의 응답시간을 향상하기 위한 데이터 마이닝 기반 데이터 패턴 분석 및 메모리 고정정책

이현섭*

백석대학교 컴퓨터공학부 교수

Data Mining Based Data Pattern Analysis and Memory Pinning Scheme to Improve Response Time on Small Mobile Devices

Hyun-Seob Lee*

Professor, Division of Computer Engineering, Baekseok University

요약 최근 소형 모바일 기기는 고성능 연산 능력과 저장장치를 갖추고 있으며, 효율적인 데이터 접근이 필수적이다. 특히, 플래시 메모리는 데이터 처리를 위해 빠른 속도를 제공하지만 비대칭 읽기 쓰기 성능과 쓰기/지우기 동작이 반복될 경우 고장이 발생할 수 있는 문제가 있다. 이러한 문제를 해결하기 위해 버퍼를 사용하고 있으나, 쓰기 성능은 개선되었지만 읽기 성능은 비대칭의 응답 지연 성능 문제가 여전히 남아있다. 본 논문에서는 이러한 문제를 해결하기 위해 빈번히 접근하는 데이터를 버퍼에 고정적으로 유지하는 방법을 제안한다. 이 방법은 데이터 마이닝을 기반으로 데이터 패턴을 분석하고, 분석된 패턴의 데이터를 유지할 수 있는 메모리 자원을 할당한다. 또한, 고정적으로 사용하는 메모리 자원은 전체 저장장치의 용량의 10% 이내로 하여 자원의 사용을 최적화하였다. 마지막으로 실험을 통해 저장 장치의 성능과 수명이 향상되는 우수성을 증명한다. 제안하는 데이터 마이닝 분석 기술과 메모리 자원 최적화 기술을 통해 소형 저장장치에 최적화된 버퍼 관리 기술의 발전에 기여할 것으로 기대한다.

주제어 : 데이터 분석, 데이터 마이닝, 접근 빈도, 데이터 패턴, 플래시 메모리

Abstract In recent years, small mobile devices have been equipped with high-performance computing power and storage, and efficient data access is essential. In particular, flash memory provides high speeds for data processing, but it suffers from asymmetric read/write performance and repeated write/erase cycles that can lead to failure. Buffers have been used to solve these problems, but while write performance has improved, read performance still suffers from asymmetric response latency. To solve these problems, this paper proposes a method to keep frequently accessed data fixed in a buffer. This method analyzes data patterns based on data mining and allocates memory resources to maintain data in the analyzed patterns. In addition, we optimize the use of resources by keeping the fixed memory resources within 10% of the total storage capacity. Finally, we demonstrate through experiments that the performance and lifetime of storage devices are improved. The proposed data mining analysis technique and memory resource optimization technique are expected to contribute to the development of buffer management techniques optimized for small storage devices.

Key Words : Data Analysis, Data Mining, Access Frequency, Data Pattern, Flash Memory

*This paper was supported by 2025 Baekseok University Research Fund

*교신저자 : 이현섭(hyunseob@bu.ac.kr)

접수일 2025년 03월 18일 수정일 2025년 04월 08일 심사완료일 2025년 04월 14일

1. 서론

현대의 소형 모바일 기기는 고성능 연산 능력과 저장 장치를 갖추고 있으며, 사용자 요구에 따라 다양한 데이터를 신속하게 처리해야 한다. 특히 스마트폰과 같은 모바일 기기는 상시 전원이 공급되며, 운영체제와 애플리케이션의 원활한 실행을 위해 저장 장치에 대한 효율적인 접근이 필수적이다. 이를 위해 빠른 접근 속도를 제공하는 플래시 메모리가 폭넓게 사용되고 있다. 그러나 플래시 메모리는 기존의 저장장치와 비교하여 빠른 속도를 제공하지만 일정 횟수 이상의 데이터 쓰기와 지우기 동작을 반복하면 메모리가 마모되는 특성과 캐시 메모리나 프로그램을 직접 실행하는 프로세서 대비 높은 읽기 및 쓰기 지연 시간을 가지고 있다[1-6]. 결과적으로, 이러한 특성은 인해 소형 모바일 기기의 수명과 시스템 성능에 큰 영향을 미친다.

이러한 속도 차이를 해결하기 위해, 빈번하게 사용되는 데이터나 미래 사용 가능성이 높은 데이터를 분석하고, 접근 빈도가 높은 데이터를 보다 빠른 속도의 버퍼에 유지하는 기술이 주목받고 있다. 이러한 기법은 캐시 메모리 및 버퍼 관리 기술을 활용하여 데이터 접근 속도를 최적화함으로써 시스템의 응답성을 향상시킨다. 일반적으로 버퍼에 있는 데이터는 전원 차단이 발생했을 경우 데이터를 읽을 가능성이 높으나, 배터리를 사용하는 모바일 기기의 경우 전원 문제가 발생하거나 전원 버튼을 끌 경우 배터리를 이용해 버퍼의 데이터를 비휘발성 저장장치에 저장할 수도 있다. 또한, 저장장치의 크기가 크지 않기 때문에 일정 크기까지 버퍼를 늘리면 성능과 수명이 크게 증가할 수 있다.

본 논문에서는 모바일 기기의 저장 장치에서 데이터 접근 패턴을 분석하고, 접근 빈도가 높은 데이터를 버퍼에 효과적으로 관리하는 방법을 설계한다. 또한, 버퍼의 크기가 미치는 성능의 영향을 분석하고 모바일 환경에 최적화된 새로운 접근 방식을 논의한다. 본 연구는 모바일 기기의 저장 장치 성능을 향상하기 위한 데이터 관리 전략을 탐색함으로써, 향후 모바일 컴퓨팅 환경에서의 저장 성능을 개선하고 기대 수명을 연장하는 데 기여할 것으로 기대한다.

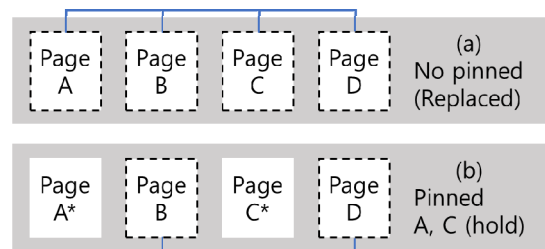
본 논문의 구성은 다음과 같다. 2장에서는 버퍼 고정 기법과 데이터 마이닝을 이용한 소형 저장장치의 데이터 패턴을 분석한다. 3장에서는 분석된 데이터 패턴을 기반으로 데이터를 메모리에 고정하는 방법과, 메모리 자원을 할당하는 방법을 논의한다. 4장에서는 실험을 통해

제안하는 기법이 저장장치의 성능과 수명연장에 미치는 영향을 증명한다. 마지막으로 5장에서 연구의 결과를 논의한다.

2. 관련연구

2.1 Buffer Pinning 기법

고정 버퍼(buffer pinning) 기법은 버퍼 관리 정책 중 한 가지 방법이다[7-9]. 이 방법은 메모리 페이지 또는 버퍼를 일정 시간 동안 해제되지 않도록 고정하는 기술이다. 이 기법은 주로 캐시 일관성을 유지하거나 특정 데이터의 빠른 접근을 보장하기 위해 사용된다. 일반적으로 운영 체제나 데이터베이스 관리 시스템의 버퍼 관리자는 메모리를 효율적으로 사용하기 위해 페이지 교체 정책을 사용한다. 그중 대표적인 방법으로 최근 사용되지 않은 페이지를 먼저 교체하는 LRU(least recently used)[10-12]와 먼저 버퍼에 적재된 데이터 순서대로 교체를 하는 FIFO(first in first out)[13, 14] 기법이 있다. 그러나 특정한 데이터가 반복적으로 접근되거나 실시간 처리가 요구되는 경우, 해당 데이터를 버퍼에 유지할 필요가 있다. 이를 위해 다른 버퍼 교체 정책보다 우선순위로 적용되는 기법이 고정 버퍼 기법이다.

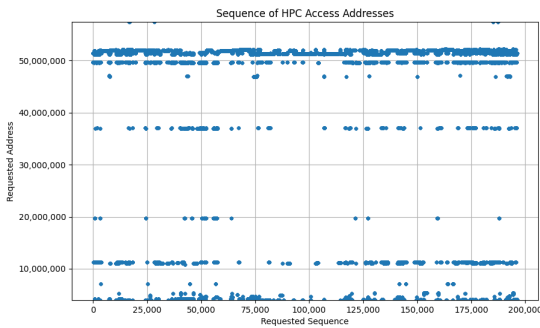


[Fig. 1] Buffer Pinning Scheme

[Fig. 1]은 고정 버퍼 기법을 적용했을 때와 적용하지 않았을 때의 차이를 보여주는 예제이다. 그림에서(a)는 No pinned의 페이지 A, B, C, D가 버퍼에 적재되어 있는 것을 보여주고 있다. 그림의 페이지들은 시스템에서 사용하고 있는 버퍼 관리 정책에 의해 교체 관리된다. 그림에서(b)는 페이지 A, C에 고정 버퍼가 적용되었다. 따라서 버퍼 교체가 발생했을 때 고정 버퍼가 적용되지 않은 페이지 B, D를 대상으로 버퍼 교체가 진행된다.

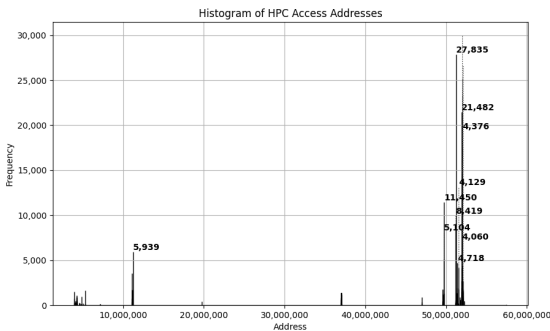
2.2 데이터 마이닝 기반 소형 저장장치 분석

빈번하게 접근하는 주소의 패턴과 범위를 발견하는 기술은 저장장치를 관리하기 위한 중요한 기술이다. 데이터 마이닝을 통해 발견된 패턴을 이용하면 저장장치의 자원 관리와 성능을 최적화하는 데 중요한 요소로 작용할 수 있다. 본 절에서는 데이터 패턴을 발굴하기 위해 고성능의 HPC(high performance computing)에서 트래이스한 주소[15]를 시각화하여 데이터 관리에 영향을 줄 수 있는 유의미한 패턴을 분석하고자 한다.



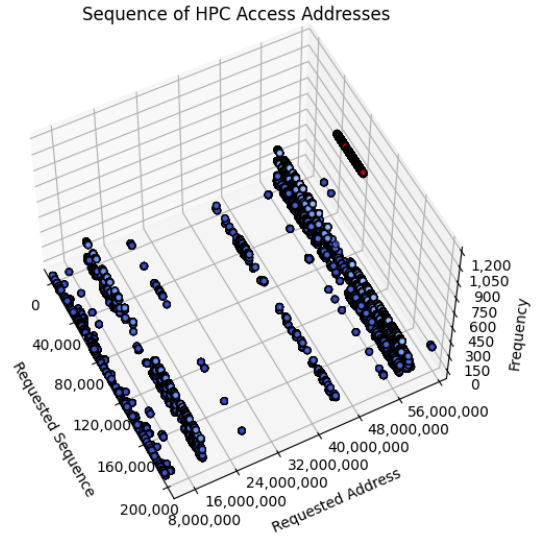
[Fig. 2] Dot Graph with Data Address

[Fig. 2]는 HPC에서 사용된 196,228개의 데이터 주소를 도트 그래프로 시각화한 결과다. 그림에서 x축은 데이터 접근 순서이고, y축은 요청된 데이터의 페이지 주소다. 도트 그래프의 각 도트는 데이터의 순서대로 접근된 데이터의 페이지 주소에 도트를 그려서 시각화하였다. 그림과 같이 특정 영역의 주소 구간의 데이터에 대한 접근이 집중되는 것을 확인할 수 있으며, 특정 시간대가 아닌 전체 시간대에 골고루 사용되는 것을 알 수 있다. 따라서, 데이터가 집중되는 구간의 주소를 분석하는 것은 저장 시스템 관리를 위해 유용할 것으로 판단된다.



[Fig. 3] Histogram with Data Address

[Fig. 3]은 HPC의 주소를 20,000개의 구간으로 나누어 히스토그램으로 분석한 결과다. 그림에서 x축은 데이터 주소의 범위고, y축은 각 주소 영역에서 접근한 데이터의 페이지 단위 횟수다. 그림에서는 4,000회 이상 접근한 데이터에만 라벨을 표시하였다. 라벨이 표시된 구간에서 최대 접근 횟수는 27,835회이고, 최저 구간은 4,060회였다.



[Fig. 4] 3D Graph based Analysis

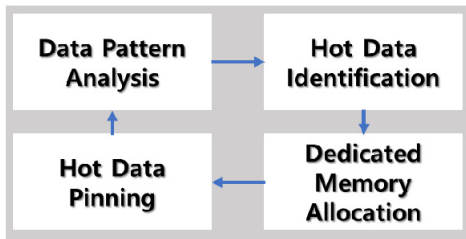
[Fig. 4]는 도트 그래프에 히스토그램 수치를 추가하여 3차원 그래프로 표현한 결과다. 그래프에서 x축은 요청된 순서고, y축은 데이터의 주소다. 그리고 z축은 각 주소 구간에서 접근한 데이터의 횟수다. 분석 결과에서 2,000회 이상 접근한 데이터의 구간은 20,000개의 구간 중 19개 구간이었고, 총 횟수는 121,564회다. 이 결과는 전체 접근한 데이터의 61.95%에 해당한다. 최소 접근 구간은 51,967,671에서 51,970,345 사이 구간의 주소고 접근 횟수는 2,026회다. 가장 높은 접근 구간은 51,264,443에서 51,267,117 구간의 주소고 접근 횟수는 27,835다. 또한 전체 주소 구간의 범위를 용량으로 약 27.20GB의 데이터이고, 이 중 2,000회 이상 접근한 구간의 주소 범위는 약 13.05MB에 해당한다. 즉, HPC에서는 0.048%의 핫데이터가 전체 저장 장치의 자원과 성능에 큰 영향을 미치고 있음을 확인할 수 있다.

분석한 결과의 같이 소형 모바일 기체에서 핫데이터에 대한 접근 빈도가 높기 때문에 수명 연장과 성능 개선을 위한 핫데이터를 관리 기술의 연구가 필요하다. 특히, 핫

데이터의 패턴에 최적화된 전용 버퍼 관리 기술은 적은 자원의 비용으로 낸드 플래시 메모리의 수명과 응답 속도를 개선할 수 있을 것으로 기대된다.

3. 소형 모바일 기기를 위한 고정 버퍼 정책

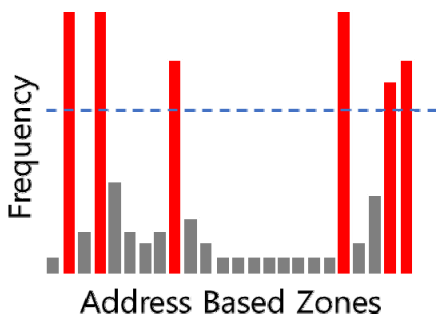
3.1 핵심 아이디어



[Fig. 5] Key Idea

Fig.5는 고정 버퍼 정책의 핵심 아이디어를 보여주고 있다. 제안하는 고정 버퍼 정책은 먼저 데이터 마이닝을 통해 데이터 패턴을 분석한다. 그다음 주소 영역별로 빈번히 접근하는 핫데이터 패턴을 식별한다. 그리고, 핫데이터를 관리하기 위한 전용 메모리 공간을 할당한다. 마지막으로, 핫데이터를 할당된 영역의 메모리 자원에 고정적으로 유지한다. 이러한 과정은 주기적으로 순환되며, 접근한 빈도수 데이터를 데이터 마이닝 하여 분석된 핫데이터를 허용할 수 있는 용량에 고정적으로 유지하여 빈번한 저장장치의 접근 횟수를 줄인다. 이 방법은 결과적으로, 긴 응답시간의 저장 미디어 접근 횟수를 줄이고 성능과 수명을 증가시킨다.

3.2 데이터 빈도수 기반 패턴 분석 정책

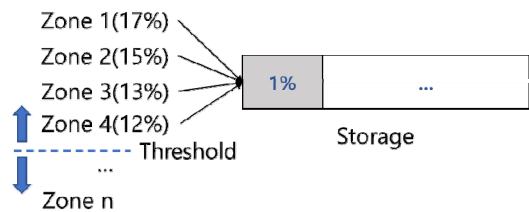


[Fig. 6] Data Pattern Analysis Policy

[Fig. 6]은 데이터 패턴 분석 정책을 보여주고 있다. 이 정책은 데이터를 n 개의 영역으로 분할하고 구간별 접근 빈도수를 측정한다. 그림의 예제에서 가로축은 주소의 구간이고 세로축은 구간별 누적된 빈도수를 의미한다. 그리고 임계선 이상의 누적된 빈도수 구간은 총 24개의 영역 중 6개의 영역이다.

3.3 핫데이터 식별 정책

핫데이터 식별 정책은 데이터의 주소를 동일한 간격으로 n 개의 영역을 생성한다. 그리고 각 영역을 영역별 누적된 접근 빈도수를 기준으로 정렬한다. 그다음 저장장치 크기의 1%의 크기 이하의 데이터 크기를 넘지 않는 범위 내에서 높은 빈도수의 영역들을 핫데이터로 식별한다. 또한, 핫데이터를 식별하기 위한 임계치의 기준은 데이터의 크기뿐만 아니라 각 영역의 누적 빈도수의 합이 전체 데이터의 누적 빈도수의 $m\%$ 이상인 경우로 정한다.



[Fig. 7] Hot Data Identification Policy

[Fig. 7]은 핫데이터 식별 정책을 보여주고 있다. 그림의 예제에서는 임계치의 값을 55%로 정하였다고 가정하였다. 이때, n 개의 영역 중 영역 1, 2, 3, 4의 크기가 전체 저장 영역의 1% 이내에 해당하고, 누적된 빈도수의 합이 임계치인 55%를 넘는다. 따라서, 영역 1, 2, 3, 4의 데이터를 핫데이터로 분류한다. 이러한 분류 기준은 데이터 마이닝 분석을 통해 전체 데이터 중 약 0.045%의 데이터에 대한 접근 횟수가 전체 데이터 접근의 약 61.95%였던 결과를 토대로 설계하였다. 즉, 핫데이터의 전체크기와 임계치 기준은 데이터 마이닝의 분석 결과를 참고하여 조정할 수 있다.

3.4 전용 메모리 할당 및 데이터 고정 버퍼 정책

이 정책은 데이터 버퍼를 관리하기 위한 정책이다. 일반적으로 버퍼 관리 정책은 LRU(least recently used) 정책을 사용한다. 그러나 전용 버퍼는 핫데이터를 고정적으로 유지하는 자원이다. 이 정책은 버퍼로 사

용하고 있는 유한 자원의 메모리에 식별된 핫데이터를 고정적으로 유지할 수 있도록 메모리 할당을 유동적으로 관리한다.

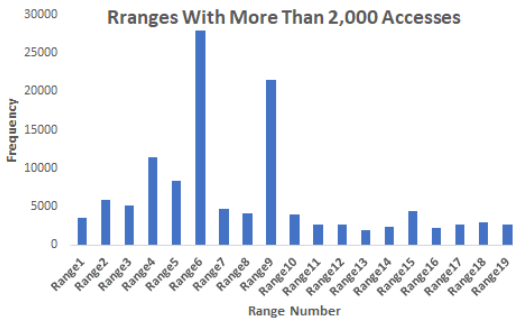


[Fig. 8] Dedicated Memory Allocation Policy

[Fig. 8]은 전용 메모리 할당 정책을 보여주고 있다. 그림의 예제에서는 버퍼 내에서 핫데이터를 유지하기 위한 메모리를 우선 배정하였고, 할당된 메모리를 데이터 고정을 위한 전용 버퍼로 사용한다. 그리고 남은 메모리는 LRU 정책으로 관리한다. 이 정책에서 핫데이터 전용 버퍼는 유동적으로 변한다. 따라서 전용 버퍼의 크기가 변경될 때마다 일반 버퍼의 데이터는 LRU 정책의 기준에 따라 데이터 적재량을 늘리거나, 가장 오랫동안 사용하지 않은 데이터를 커밋하여 버퍼의 메모리 크기를 조절한다.

4. 실험 및 결과

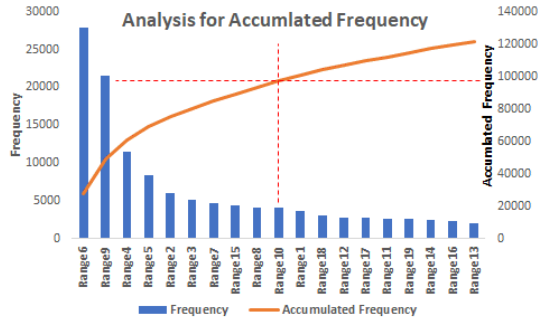
본 절에서는 데이터 마이닝을 통해 식별된 핫데이터를 고정 버퍼에 유지하는 것이 읽기 요청에 대한 응답시간에 미치는 영향을 분석하기 위해 실험을 통한 분석을 수행하였다. 실험에 사용된 트레이스는 196,288개였다.



[Fig. 9] Frequency Analysis per Zone

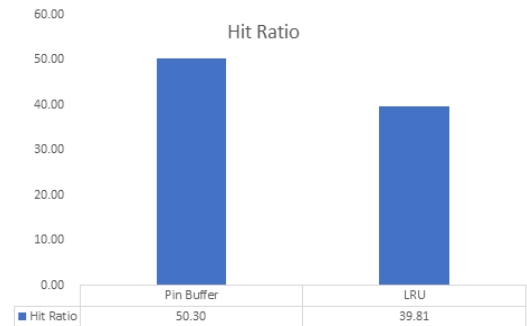
[Fig. 9]는 트레이스를 주소별로 분할하고, 2,000회 이상 접근한 주소 영역을 시각화한 그래프다. 전체 2만

개의 영역 중 19개의 영역이 2,000회 이상 사용되었고 전체 트레이스의 약 61.93%였다.



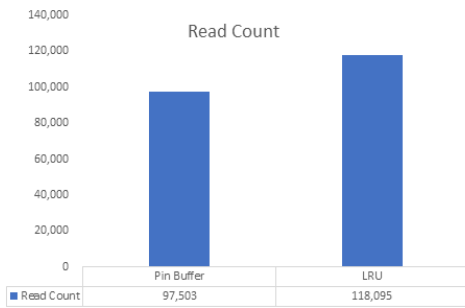
[Fig. 10] Visualization for Threshold Analysis

[Fig. 10]은 메모리 자원 할당을 위해 임계값 분석을 위한 시각화의 결과이다. 그림의 그래프는 Fig. 9의 결과를 접근 빈도를 기준으로 재배열하고, y축에 빈도수뿐만 아니라 누적 빈도를 추가한 것이다. 실험의 시뮬레이션은 메모리 자원은 전체 트레이스의 약 50%인 98,114회를 감당할 수 있도록 설계했고, 이것은 상위 빈도수 기준 10개 범위의 영역에서 발생한 누적 빈도수는 97,512회로 50%에 가장 근접한 양의 데이터가 처리되었다.



[Fig. 11] Hit Ratio

[Fig. 11]은 고정 버퍼 기법의 Hit ratio를 보여주고 있다. 실험에서 분석된 데이터를 기반으로 고정 버퍼의 크기는 5MB로 하였고, 1MB는 LRU 기법을 적용하였다. 반면, 실험의 비교군은 전체 6MB를 LRU로 할당하였다. 실험 결과와 같이, 고정 버퍼 기법과 LRU의 Hit ratio는 각각 39.81%, 50.30%로 LRU와 비교하여 약 10.49% 높았다.



[Fig. 12] Read Count

[Fig. 12]는 읽기 횟수 결과를 보여주고 있다. 그림과 같이 고정 버퍼 기법과 LRU의 읽기 횟수는 각각 97,503 회, 118,095회 였고, 고정 버퍼 기법의 성능이 약 17.44% 높았다. 실험을 통해 제안하는 기법이 이전에 널리 사용되고 있는 LRU 기법보다 더 우수한 성능을 보이는 것을 확인할 수 있었다.

5. 결론

본 논문에서는 데이터 마이닝을 통해 빈번히 접근하는 주소 핫데이터 영역을 분석하였고, 핫데이터 영역을 고정 버퍼 기법으로 관리하였을 때 LRU보다 더 우수한 응답 성능을 보이는 것을 확인하였다. 향후에는 고정 버퍼의 적재율을 높이고 성능을 향상하는 다양한 방법을 연구할 예정이다.

REFERENCES

[1] H.S.Lee, "A Design of SSD Dedicated RAID System for Efficient Resource Management," *Journal of Internet of Things and Convergence*, Vol.10, No.2, pp.109-114, 2024.

[2] H.S.Lee, "An Efficient Resource Optimization Method for Provisioning on Flash Memory-Based Storage," *Journal of Internet of Things and Convergence*, Vol.9, No.4, pp.9-14, 2023.

[3] H.S.Lee, "A Memory Mapping Technique to Reduce Data Retrieval Cost in the Storage Consisting of Multi Memories," *Journal of Internet of Things and Convergence*, Vol.9, No.1, pp.19-24, 2023.

[4] H.S.Lee, "Performance analysis and prediction through various over-provision on NAND flash memory based storage," *Journal of Digital Convergence*, Vol.20, No.3, pp.343-348, 2022.

[5] H.S.Lee, "A Design of Temperature Management System for Preventing High Temperature Failures on Mobility Dedicated Storage," *Journal of Internet of Things and Convergence*, Vol.10, No.2, pp.125-130, 2024.

[6] H.S.Lee, "An Efficient SLC Transition Method for Improving Defect Rate and Longer Lifetime on Flash Memory," *Journal of Internet of Things and Convergence*, Vol.9, No.3, pp.81-86, 2023.

[7] J.Kim, J.Y.Lee, J.I.Park, D.B.Min and K.H.Sohn, "Pin the Memory: Learning To Generalize Semantic Segmentation," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4350-4360, 2022.

[8] N.Riekenbrauck, M.Weisgut, D.Lindner and T.Rabl, "A Three-Tier Buffer Manager Integrating CXL Device Memory for Database Systems," *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pp.395-401, 2024.

[9] J.Lee and D.Kim, "Large-scale 3D fast Fourier transform computation on a GPU," *ETRI Journal*, Vol.45, No.6, pp.1035-1045, 2023.

[10] J.Yang, Y.Wang and Z.Wang, "Efficient Modeling of Random Sampling-Based LRU," *ICPP '21: Proceedings of the 50th International Conference on Parallel Processing*, No.32, pp.1-11, 2021.

[11] Q.Zhao and Q.Ran, "A Buffer Algorithm of Flash Database Based on LRU," *International Symposium on Computing and Artificial Intelligence (ISCAI)*, pp.120-124, 2022.

[12] Y.B.Mazziane, S.Alouf, G.Neglia and D.S.Menasche, "TTL model for an LRU-based similarity caching policy," *Computer Networks*, Vol.241, 2024.

[13] V.Panchbhaiyye and T.Ogunfunmi, "An Efficient FIFO Based Accelerator for Convolutional Neural Networks," *Journal of Signal Processing Systems*, Vol.93, pp.1117-1129, 2021.

[14] E.A.Aksenova and A.V.Sokolov, "Optimal Parallel Control of n FIFO-Queues in Shared Memory," *Journal of Mathematics*, Vol.42, pp.44-49, 2021.

[15] SNIA, <http://iotta.snia.org/traces/hpc>.

이 현 섭(Hyun-Seob Lee)

[종신회원]



- 2013년 2월 : 한양대학교 컴퓨터 공학과 (공학 박사)
- 2012년 3월 ~ 2021년 2월 : 삼성전자 책임연구원
- 2021년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 조교수

<관심분야>

인공지능, 저장시스템, 임베디드 시스템