사물인터넷융복합논문지 Vol. 9, No. 5, pp. 11-16, 2023 https://doi.org/10.20465/KIOTS.2023.9.5.011

효율적인 자원 관리를 위한 핫데이터 분류기 및 관리 정책 설계

이현섭*
백석대학교 컴퓨터공학부 교수

A Design of Hot Data Classifier and Management Scheme for An Efficient Resource Management

Hyun-Seob Lee*
Professor, Division of Computer Engineering, Baekseok University

요 약 핫데이터는 저장시스템의 성능과 수명에 영향을 주는 사용 빈도가 높은 데이터이다. 따라서 식별된 핫데이터를 분류하여 관리하는 것은 저장장치의 성능과 수명을 위해 중요한 일이다. 특히 특정 블록에 집중적인 쓰기 동작을 반복하면 열화 현상이 급격하게 진행되는 특징을 가지고 있는 낸드 플래시 메모리는 이러한 핫데이터를 구분하여 관리하는 것이 성능과 수명을 향상하기 위해 필요하다. 본 논문에서는 엔터프라이즈 서버에서 추출한 트레이스를 벤치마킹하여 플래시 메모리 기반 저장장치에서 자원을 효율적으로 관리하는데 필요한 핫데이터 분류기 및 관리 정책을 설계한다. 또한. 멀티 해시를 이용하여 핫데이터의 빈도를 측정하고, 식별된 핫데이터를 구분하여 열화에 강한 미디어에 관리하는 기법을 통해 저장시스템의 성능과 수명을 향상시키는 방법을 제안한다. 마지막으로 실험을 통해 증가한 성능과 수명을 측정하여 제안하는 방법의 효과를 증명한다.

주제어 : 낸드 플래시 메모리, 멀티 해시, 핫데이터, 데이터 식별, 데이터 분류

Abstract Hot data is frequently used data that affects the performance and lifespan of a storage system. Therefore, classifying and managing the identified hot data is important for the performance and lifetime of storage devices, especially for NAND flash memory, which is characterized by rapid degradation after repeated intensive write operations to specific blocks. In this paper, we design a hot data classifier and management scheme for an efficient resource management in flash memory-based storage devices by benchmarking traces extracted from enterprise servers. Also, we propose a method to measure the frequency of hot data using multiple hashes, and to improve the performance and lifetime of the storage system by classifying the identified hot data and managing it on degradation-resistant media. Finally, we demonstrate the effectiveness of the proposed method by measuring the increased performance and lifetime through experiments.

Key Words : Nand Flash Memory, Multi-Hash, Hot Data, Data Identification, Data Classification

*This paper was supported by 2023 Baekseok University Research Fund
*교신저자 : 이현섭(hyunseob@bu.ac.kr)
접수일 2023년 8월 8일 수정일 2023년 10월 6일 심사완료일 2023년 10월 9일

1. 서론

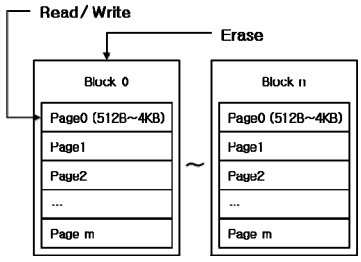
핫데이터(hot data)[1, 2]는 저장 시스템에서 사용되는 전체 데이터 중 1% 미만의 작은 용량을 차지한다. 그러나 데이터 사용 빈도가 높아서 저장장치의 성능과 수명에 영향을 주는 특징을 가지고 있다. 따라서 대용량의 빅데이터(big data)를 관리하는 엔터프라이즈 및 데이터 센터에서는 이러한 핫데이터를 분류하고 관리하는 것이 시스템의 전체적인 성능과 수명을 위해 중요한 일이다. 특히 저전력, 고집적도와 빠른 읽기 쓰기 성능때문에 엔터프라이즈의 주요 저장장치로 활용되고 있는 SSD (solid state disk)는 낸드 플래시 메모리의 구성과 물리적 특징 때문에 핫데이터에 의한 영향 크게 받는다. 따라서 SSD의 성능 향상과 수명 효율을 개선하기 위해서는 핫데이터를 관리하는 정책이 필요하다.

플래시 메모리의 구성은 여러 개의 페이지를 유지하고 있는 다수의 블록으로 이루어져 있다. 그리고 데이터를 읽고 쓰는 단위는 페이지 단위로 처리되는 반면 데이터를 지우는 단위는 블록 단위로 처리되는 특징이 있다. 플래시 메모리의 또 한가지 독특한 물리적 동작 특성은 데이터가 쓰이지 않은 페이지에만 데이터를 쓸 수 있다는 것이다. 따라서 데이터가 쓰여 있는 페이지에 업데이트가 발생하면 데이터를 쓰기 전 해당 블록이 지우고 써야 하며 이 경우 블록 내에 데이터를 유지하고 있는 모든 페이지의 데이터를 다른 블록으로 백업(backup)하는 동작이 선행돼야 한다. 이러한 특징을 쓰기 전 지우기 (erase-before-write)라고 한다. 즉, 이러한 특징 때문에 중복된 페이지에 쓰기 요청이 처리될 경우 단일 쓰기 동작만으로 데이터를 처리할 수 없고, 다수 페이지의 읽기 쓰기 동작과 블록 단위의 지우기 동작이 선행돼야 하는 문제가 있다[3-6]. FTL(flash translation layer)은 논리적인 주소를 물리적인 주소로 매핑하여 이러한 특징 때문에 발생하는 문제를 완화해 준다. 그러나 핫데이터와 같이 작은 영역에 집중적인 쓰기 요청이 처리되는 경우 플래시 메모리의 근원적인 특징으로 발생하는 성능저하와 수명저하 문제는 피할 수 없다. 따라서 최근에는 핫데이터를 분리하기 위한 연구와 핫데이터를 관리하기 위한 연구가 다양하게 진행되고 있다[7-11].

본 논문에서는 엔터프라이즈 서버에서 추출한 트레이스를 벤치마킹하여 핫데이터의 분포와 특징을 분석한다. 그리고 분석된 벤치마크를 기반으로 자원을 효율적으로 관리하는데 필요한 핫데이터 분류기를 설계한다. 또한 실험을 통해 식별된 핫 데이터를 열화에 강한 단일 비트 블록에 관리하는 기법을 적용하여 증가한 성능과 수명 연장 효과를 증명한다.

2. 배경

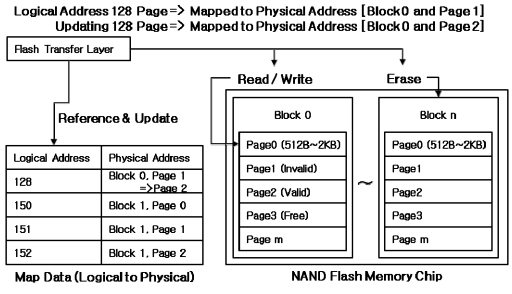
2.1 낸드 플래시 메모리의 특징



[Fig. 1] NAND Flash Memory

Fig. 1은 n개의 블록으로 구성된 플래시 메모리의 구성을 보여주고 있다. 그림에서 각 블록은 데이터의 기본 읽기 쓰기 단위인 페이지를 m개 유지하고 있다. 플래시 메모리의 대표적인 성능 특징은 읽기 쓰기 지우기 동작이 데이터 처리 단위뿐만 아니라 처리 성능 또한 비대칭이라는 것이다. 일반적으로 읽기 쓰기 지우기 성능 비율은 1:10:100이다. 따라서 성능향상을 위해 읽기 및 쓰기 연산이 증가하더라도 지우기 연산을 지연시키는 동작이 필요하다[12-14].

2.2 낸드 플래시 관리 기법



[Fig. 2] Role of FTL

Fig. 2는 FTL의 역할을 보여주고 있다. 그림과 같이 FTL은 논리적인 주소를 물리적인 주소로 전환해 준다. 그림의 예에서 논리적인 주소 128은 물리적인 주소 0번 블록 1번 페이지에 저장되어 있었다. 그리고 128번 주소에 업데이트가 발생하여 데이터를 다시 써야 하는 상황

이 발생했다. FTL은 0번 블록 내에 비어있는 2번 페이지에 데이터를 쓰도록 자원을 할당하고 매핑 테이블을 수정하였다. 이러한 동작을 통해 FTL은 블록 내에 모든 페이지가 소진될 때까지 지우기 동작을 지연시킬 수 있다. 그러나 특정 영역에 집중적인 쓰기 요청이 발생하면 FTL의 알고리즘만으로는 성능저하 및 불량 발생을 피하기 어렵다. 특히 플래시 메모리는 데이터를 유지하는 셀의 타입에 따라 약 1만회에서 10만회 반복 쓰고 지우기 동작을 수행했을 때 마모(wear out)로 인한 고장이 발생한다. 따라서 집중적인 쓰기 요청을 발생시키는 핫데이터를 관리하기 위한 추가적인 기술이 필요하다.

3. 효율적인 핫데이터 분류기 설계

3.1 데이터 분석 및 벤치마크

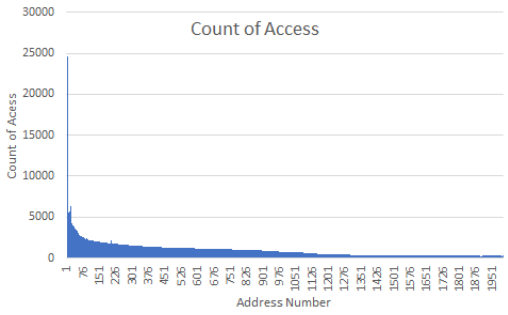
본 절에서는 핫데이터 분류기 설계의 정확성과 효과성을 위해 마이크로소프트사의 엔터프라이즈 서버에서 약 1주일간 수집된 실제 트레이스[15]를 분석하여 벤치마크했다. 수집된 트레이스에서 저장장치의 페이지 크기는 512B이고 약 350MB 영역에서 718,703개의 페이지에 492,892회 쓰기 요청을 수행하였다.

〈Table 1〉 Hot Data Analysis

Volume	350MB	3.5MB	1MB	350KB
Count	5,325,660	2,663,817	1,875,309	1,224,225
Transfer	2.54GB	1.27GB	0.916GB	0.598GB

Table 1은 실험에서 사용된 데이터의 핫데이터를 분석한 결과이다. 표에서 보여주는 것과 같이 약 350MB 영역에 중복된 페이지를 포함하여 5,325,660개의 페이지로 2.54GB의 쓰기 데이터를 전송하였다. 그리고 1% 영역인 3.5MB에 전체 쓰기 데이터의 50%인 1.27GB를 전송하였다. 1MB 영역에는 36.06%의 데이터를 전송하였고, 0.1% 영역인 350KB에는 전체 데이터의 23.54% 전송하였다. 분석 결과를 통해 전체 저장 영역 중 앞 영역에 쓰기 연산의 빈도가 높은 핫데이터가 밀집해 있는 것을 확인할 수 있다.

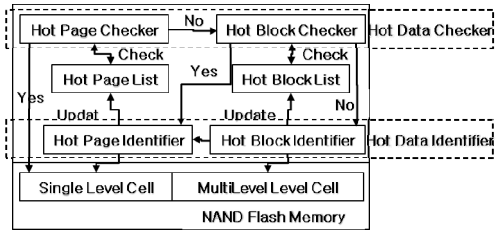
Fig. 3은 전체 데이터 중 약 36%의 쓰기 요청이 발생한 1MB 영역에서 각 페이지 주소별 쓰기 요청 횟수를 보여주고 있다. 그림에서 X축은 페이지 주소를 의미하고, Y축은 페이지별 발생한 쓰기 요청의 접근 횟수이다.



[Fig. 3] Count of Access

그림과 같이 전체 영역 중 앞 영역에 많은 횟수의 쓰기 요청이 있었고 특히 50% 영역에 해당하는 512KB 영역에 약 78.38%의 쓰기 요청이 발생하였다. 이중 가장 적은 요청 횟수는 마지막 페이지에 발생한 753회로 약 376KB의 데이터 쓰기 요청을 수행했다. 이를 기반으로 512KB를 수용할 수 있는 용량의 블록을 단일 비트 블록으로 할당하고 핫데이터를 관리하기 위한 전용 공간으로 사용한다.

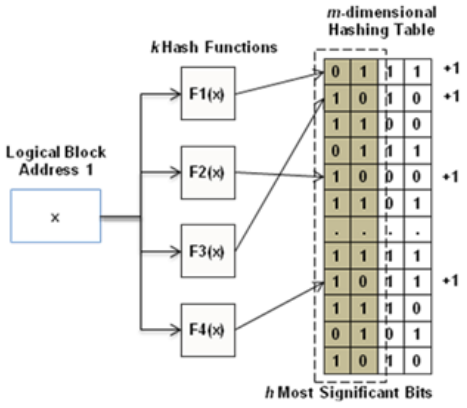
3.2 핫데이터 분류기 및 SLC 전환기법의 설계



[Fig. 4] Key Idea

Fig. 4는 핫데이터 분류기를 이용한 SLC 전환기법의 핵심 아이디어를 보여주고 있다. 제안하는 기법은 해시 기반의 식별기를 이용하여 핫데이터를 식별한다. 식별된 핫데이터는 반일 비트 블록에서 관리한다. 그리고 콜드 데이터는 로그 기반 멥핑 방법을 이용하여 멀티 비트 블록에서 관리한다. 핫데이터를 식별하는 방법은 그림의 예제와 같이 논리적 주소에 대한 데이터 처리가 발생했을 때 학습된 리스트로부터 핫섹터로 학습되었는지 확인을 한다. 만약 핫섹터로 확인되면 섹터 멥핑으로 데이터를 관리한다. 핫섹터가 아닌 경우 핫블록인지 여부를 다시 한번 확인한다. 만약 핫블록인 경우 핫섹터인지 다시 한번 학습한다. 그 밖에 다른 데이터는 핫블록 인식을 위한 학습을 수행 후 블록 멥핑 기법을 사용하여 관리한다.

3.3 해시 기반 핫데이터 분류기

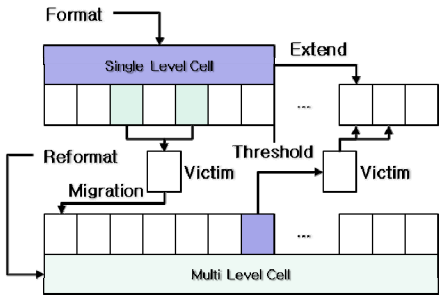


[Fig. 5] Hash Based Hot Data Identifier

Fig. 5는 해시 기반 핫데이터 분류기의 설계를 보여주고 있다. 이 방법은 k개의 해시함수와 하나의 엔트리당 m개의 비트버킷으로 이루어진 해시테이블로 구성한다. 이 예제에서 k와 m은 각각 4이다. 그리고 논리적 주소에 대해 k개의 해시 함수를 통해 각각의 해시 값을 산출하고 해시 값에 해당하는 해시테이블의 버킷에 1씩 값을 누적한다. 각 비트버킷은 h개의 중요 비트와 일반 비트로 구분하는데, 그림에 예제에서는 2개의 비트를 중요 비트로 설정하였다. 중요 비트는 논리주소에 대해 해시의 결과가 모두 중요 비트영역에 적중되었을 때 핫데이터로 처리한다. 그리고 이 테이블의 버킷은 일정 주기마다 우측으로 비트를 시프트 하는 방법을 사용한다. 이 방법은 적은 오버헤드로 핫데이터를 식별할 수 있는 장점이 있다. 대용량의 데이터에서 핫데이터를 식별하는 것은 비트버킷의 길이와 개수에 민감한 영향을 받는다. 따라서 실험을 위한 상세 설계는 벤치마크를 기반으로 약 0.1%의 핫데이터를 식별할 수 있는 기준 설정을 사용하였다. 0.1%인 350KB 구간 내 최저 접근 횟수는 1085였고, 500KB 구간 내 최저 접근 횟수는 748이였다. 따라서 비트버킷의 크기는 1,023회 이하를 구분할 수 있는 10 비트로 설정하였다. 이 설정은 전체 구간 중 약 0.14%를 식별할 수 있다.

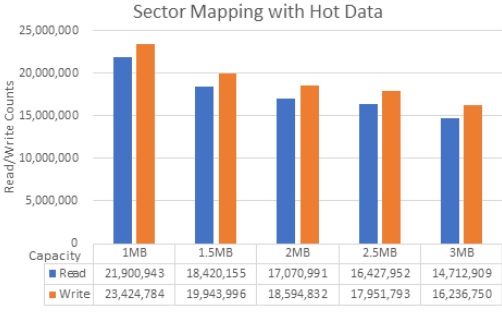
3.4 단일 레벨 셀 블록 전환 기법

Fig. 6은 단일 레벨 셀 블록 전환 기법의 핵심 설계를 보여주고 있다. 이 기법은 플래시 메모리를 단일 비트 셀과 다중 비트 셀로 포맷하고 데이터의 사용 빈도에 따라 다른 포맷의 메모리로 데이터를 관리한다. 그림의 예에



[Fig. 6] SLC Translation Method

서는 7개의 블록을 단일 레벨 셀로 포맷했다. 그리고 최초의 데이터는 단일 레벨 셀 블록에 관리하고 사용량이 많지 않으면 멀티 레벨 셀로 포맷된 블록으로 이동시킨다. 반면 멀티 레벨 셀 블록에 저장된 데이터 중 사용 빈도가 높은 데이터는 단일 레벨 셀의 블록으로 이동시키며 핫데이터의 사용량에 따라 블록 포맷을 통해 단일 레벨 셀 블록과 멀티 레벨 블록 셀의 임계치를 조절한다. 이 방법은 사용 빈도가 높은 핫데이터를 내구성이 좋은 단일 비트 셀 블록에 관리함으로써 성능과 수명을 향상하는 장점이 있다. 그러나 몇 개의 블록을 단일 비트 블록으로 설정할지 결정하는 것은 안정화가 되기까지 큰 비용이 발생한다. 따라서 벤치마크를 기반으로 실험하여 최적화된 자원을 결정하였다.



Capacity	1MB	1.5MB	2MB	2.5MB	3MB
Read	21,900,943	18,420,155	17,070,991	16,427,952	14,712,909
Write	23,424,784	19,943,996	18,594,832	17,951,793	16,236,750

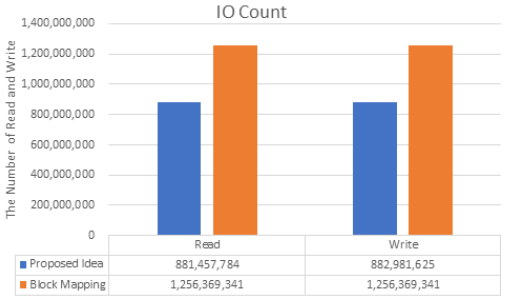
[Fig. 7] Sector Mapping with Hot Data

Fig. 7은 벤치마크의 핫데이터를 섹터 매핑 기법으로 관리했을 때 발생한 읽기/쓰기 횟수이다. 실험에서 사용된 벤치마크의 핫데이터 영역은 0에서 512KB 영역이고 트레이스 분석결과 약 0.718GB 데이터에 대한 쓰기 요청이 발생하였다. 그림의 결과와 같이 자원이 증가할수록 접근 횟수가 감소하는 경향을 보였고, 자원이 3MB일 때 1MB보다 읽기/쓰기 성능이 각각 32.82%와 30.68% 증가하였다. 따라서 핫데이터 전용 블록은 3MB를 유지

할 수 있는 24개의 블록을 사용하였다.

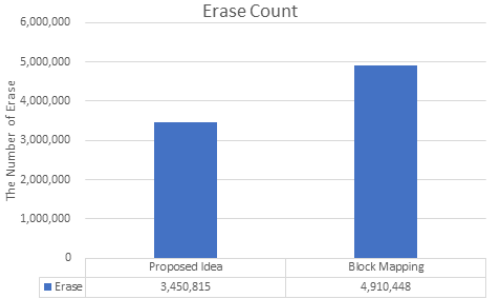
4. 실험 및 평가

제안하는 아이디어의 성능을 검증하기 위해 400GB 크기의 저장장치 환경에서 시뮬레이션을 수행하였다. 실험에서 페이지의 크기는 512B이고 각 블록당 페이지의 개수는 256개로 설정하였다. 또한, 단일 레벨 셀 블록은 앞선 실험을 통해 3MB로 설정하였다. 그리고 벤치마크에서 약 350MB 저장공간에 약 2.54GB의 쓰기 데이터를 요청하는 트레이스를 사용하였다. 마지막으로 제안하는 아이디어와 기본적인 동작을 하는 블록 매핑 기법을 각각 시뮬레이션하여 벤치마크 데이터의 트레이스를 수행하는 동한 발생하는 성능을 비교 분석하였다.



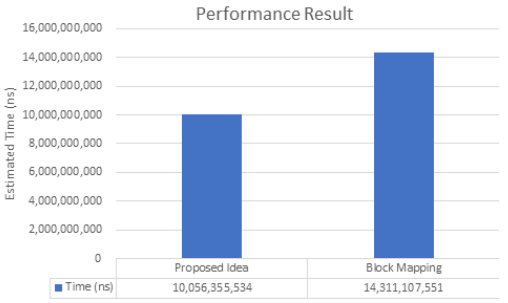
[Fig. 8] The Number of Read and Write

Fig. 8은 벤치마크 트레이스에 따라 요청된 데이터의 쓰기 동작을 수행하는 동안 발생한 읽기/쓰기 횟수를 보여주고 있다. 읽기 횟수는 각각 881,457,784, 1,256,369,341회로 제안하는 아이디어가 약 29.84% 우수한 성능을 보였다. 또한, 쓰기 횟수도 약 29.72% 우수한 성능을 보였다.



[Fig. 9] The Number of Erase

Fig. 9는 벤치마크를 수행하는 동안 내부적으로 발생한 지우기 동작의 횟수를 보여주고 있다. 제안하는 아이디어와 기본적인 블록 매핑 기법이 각각 3,450,815, 4,910,448회의 지우기 연산을 수행하다. 결과적으로 실험에서 제안하는 방법이 약 29.73% 우수한 지우기 연산 성능을 보였다.



[Fig. 10] Estimated Time

Fig. 10은 벤치마크를 수행하는 동안 발생한 읽기/쓰기/지우기 횟수를 기반으로 소모된 시간을 ns 단위로 시뮬레이션한 결과이다. 실험에서는 각각 10,056,355,534, 14,311,107,551ns의 시간이 소모되었고 제안하는 방법이 블록 매핑 기법과 비교하여 약 29.73% 우수한 성능을 보였다.

5. 결론

본 논문에서는 플래시 메모리 기반 저장장치에서 자원을 효율적으로 관리하기 위한 핫데이터 분류기 및 단일 비트 셀 블록 관리 정책을 설계하였다. 설계한 방법은 멀티 해시를 이용하여 핫데이터를 구분한다. 그리고 식별된 핫데이터를 열화에 강한 단일 비트 셀 블록에 관리하는 기법을 통해 저장시스템의 성능과 수명을 증가하는 방법을 제안하였다. 또한, 엔터프라이즈 서버의 트레이스를 벤치마킹하여 자원을 최적화하였다. 마지막으로 실험을 통해 제안하는 아이디어의 성능 향상을 증명하였다. 앞으로는 핫데이터를 식별하고 관리하기 위한 자원을 데이터 패턴에 따라 적응형으로 관리하기 위한 연구와 핫데이터 관리 기법을 통한 수명 증가 기법을 연구할 예정이다.

REFERENCES

[1] Q.Luo, X.Fang, Y.Sun, J.Ai and C.Yang, "Self-learning hot data prediction: Where echo state network meets NAND flash memories," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.67, No.3, pp.939-950, 2020.

[2] F.Yu, and H.Yan, "An Efficient Hot-Cold Data Separation Garbage Collection Algorithm Based on Logical Interval in NAND Flash-based Consumer Electronics," *IEEE Transactions on Consumer Electronics,* Vol.69, No.3, pp.431-440, 2023.

[3] H.S.Lee, "A method for optimizing lifetime prediction of a storage device using the frequency of occurrence of defects in NAND flash memory," *Journal of Internet of Things and Convergence,* Vol.7, No.4, pp.9-14, 2021.

[4] H.S.Lee, "High Efficiency Life Prediction and Exception Processing Method of NAND Flash Memory-based Storage using Gradient Descent Method," *Journal of Convergence for Information Technology*, Vol.11, No.11, pp.44-50, 2021

[5] H.S.Lee, "A Safety IO Throttling Method Inducting Differential End of Life to Improving the Reliability of Big Data Maintenance in the SSD based RAID," *Journal of Digital Convergence,* Vol.20, No.5, pp.593-598, 2022.

[6] 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.

[7] S.Yoo, and D.Shin, "Reinforcement Learning-Based SLC Cache Technique for Enhancing SSD Write Performance," *In 12th USENIX Workshop on Hot Topics in Storage and File Systems*, 2020.

[8] S.Y.Lim and H.J.Chang, "Airflow management analysis to suppress data center hot spots," *Building and Environment*, Vol.197, pp.107843, 2021.

[9] X.Ye and S.Xu, "Study on surface defect classification of hot-rolled strip based on PSO-SVM," *In Proceedings of the Eighth Asia International Symposium on Mechatronics*, Vol.885, pp.1846-1855, 2021.

[10] S.Rajendran, O.I.Khalaf, Y.Alotaibi and S.Alghamdi, "MapReduce-based big data classification model using feature subset selection and hyperparameter tuned deep belief network," *Scientific Reports*, Vol.11, No.1, pp.24138, 2021.

[11] S.Wu, W.Zhang, B.Mao and H.Jiang, "HotR: Alleviating Read/Write Interference with Hot Read Data Replication for Flash Storage," *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp.1367-1372, 2019.

[12] H.S.Lee, "A Prediction-Based Data Read Ahead Policy using Decision Tree for improving the performance of NAND flash memory based storage devices," *Journal of Internet of Things and Convergence,* Vol.8, No.4, pp.9-15, 2022.

[13] 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

[14] H.S.Lee, "A Study on the Performance Measurement and Analysis on the Virtual Memory based FTL Policy through the Changing Map Data Resource," *Journal of Internet of Things and Convergence,* Vol.9, No.1, pp.71-76, 2023

[15] *http://iotta.snia.org/traces/block-io/388*

이 현 섭(Hyun-Seob Lee) [종신회원]

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

〈관심분야〉

인공지능, 저장시스템, 임베디드 시스템