

플래시 메모리 기반 저장장치에서 프로비저닝을 위한 효율적인 자원 최적화 기법

이현섭*

백석대학교 컴퓨터공학부 교수

An Efficient Resource Optimization Method for Provisioning on Flash Memory-Based Storage

Hyun-Seob Lee*

Professor, Division of Computer Engineering, Baekseok University

요약 최근 엔터프라이즈 및 데이터 센터에서는 급격하게 증가하고 있는 빅데이터를 관리하기 위한 자원 최적화 연구가 활발하게 진행되고 있다. 특히 고정 할당된 저장 자원과 비교하여 많은 자원을 할당하는 씬프로비저닝은 초기 비용을 줄이는 효과가 있으나 실제로 사용하는 자원이 증가할수록 비용의 효과는 감소하고 자원을 할당하기 위한 관리 비용이 증가하는 문제가 있다. 본 논문에서는 플래시 메모리의 물리적 블록을 단일 비트 셀과 다중 비트 셀로 구분하여 하이브리드 기법으로 포맷하고, 빈번하게 사용하는 핫 데이터와 사용량이 적은 콜드 데이터를 구분하여 관리하는 기법을 제안한다. 제안하는 기법은 씬프로비저닝과 같이 물리적으로 자원과 할당된 자원이 동일하여 추가적인 비용 증가 없이 사용할 수 있으며, 사용량이 적은 자원을 다중 비트 셀 블록에 관리하여 씬프로비저닝과 같이 일반적인 저장장치보다 더 많은 자원을 할당할 수 있는 장점이 있다. 마지막으로 시뮬레이션을 기반으로 실험을 통해 제안하는 기법의 자원 최적화 효과를 측정하였다.

주제어 : 낸드 플래시 메모리, 다중 비트 셀, 단일 비트 셀, 씬프로비저닝, 씬프로비저닝

Abstract Recently, resource optimization research has been actively conducted in enterprises and data centers to manage the rapid growth of big data. In particular, thin provisioning, which allocates a large number of resources compared to fixedly allocated storage resources, has the effect of reducing initial costs, but as the number of resources actually used increases, the cost effectiveness decreases and the management cost for allocating resources increases. In this paper, we propose a technique that divides the physical blocks of flash memory into single-bit cells and multi-bit cells, formats them with a hybrid technique, and manages them by dividing frequently used hot data and infrequently used cold data. The proposed technique has the advantage that the physical and allocated resources are the same, such as thick provisioning, and can be used without additional cost increase, and the underutilized resources can be managed in multi-bit cell blocks, such as thin provisioning, which can allocate more resources than typical storage devices. Finally, we estimated the resource optimization effectiveness of the proposed technique through experiments based on simulations.

Key Words : nand flash memory, multi level cell, single level cell, thin provisioning, thick provisioning

*This paper was supported by 2023 Baekseok University Research Fund

*교신저자 : 이현섭(hyunseob@bu.ac.kr)

접수일 2023년 6월 30일 수정일 2023년 8월 4일 심사완료일 2023년 8월 7일

1. 서론

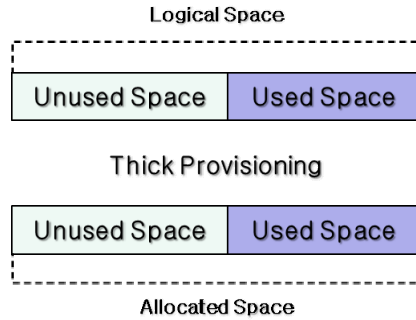
최근 엔터프라이즈 및 데이터 센터에서는 급격하게 증가하고 있는 빅데이터(big data)를 관리하기 위한 자원 최적화 연구가 활발하게 진행되고 있다. 그러나 빅데이터를 관리하기 위한 저장 시스템의 자원 증가 속도는 급격하게 증가하고 있는 데이터의 속도를 수용하는 데 어려움이 있다. 이러한 문제를 극복하고 저장 시스템의 자원 비용을 줄이기 위해 씬프로비저닝(thin provisioning) 기법이 연구되었다. 이 방법은 저장 시스템의 물리적인 자원보다 사용자에게 보여주는 가상의 자원이 더 큰 비대칭 자원 할당 방법으로 물리적 자원을 구축하기 위한 초기 비용을 줄이는 장점이 있다. 반면 실제 데이터의 양이 증가할 경우 물리적 자원의 비용 감소 효과는 줄어들고 데이터를 위해 자원을 할당하기 위한 관리 비용이 추가로 발생하는 문제가 있다. 본 논문에서는 데이터 증가로 인해 발생하는 문제를 해결하기 위해 저장 시스템의 초기부터 빈번하게 사용되는 핫 데이터와 비정규적으로 사용되는 콜드 데이터를 구분하여 관리하기 위한 하이브리드 포맷 기법을 제안한다. 이 기법은 빈번하게 사용되는 핫 데이터는 물리적 특성이 좋은 단일 비트 셀에서 관리하고, 사용량이 적은 콜드 데이터는 데이터를 유지할 수 있는 용량에 장점이 있는 다중 비트 셀 블록에 관리한다. 따라서 데이터가 증가에 따라 추가해야 하는 물리적 비용을 감소시키는 효과가 있다. 또한, 가상의 자원과 물리적 자원을 동일 크기의 용량으로 관리하여 비대칭 자원 크기 때문에 발생하는 자원 관리 비용을 줄이는 효과가 있다. 마지막으로 시뮬레이션을 통해 제안하는 기법의 자원 최적화 효과를 측정한다.

2. 배경

2.1 씬프로비저닝 (Thin Provisioning)

엔터프라이즈 환경에서 저장 시스템의 자원은 사용자에게 보여주는 논리적 자원과 시스템 내부의 물리적 자원으로 구분된다. 그리고 자원 관리를 위한 전통적인 방법은 씬프로비저닝이다[1].

Fig. 1은 씬프로비저닝을 보여주고 있다. 그림에서 사용자에게 보여주는 논리적 공간은 실제 데이터를 저장하기 위해 사용된 공간과 사용되지 않은 공간이 있다. 그리고 씬프로비저닝은 실제 사용된 공간과 관계없이 논리적 공간크기의 자원을 물리적 공간에 미리 할당한다. 이 방

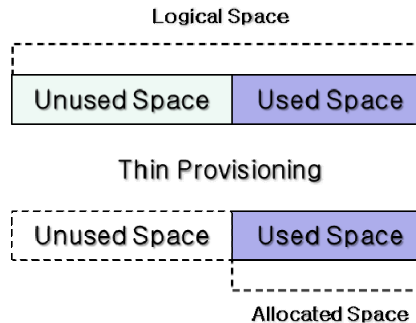


[Fig. 1] Thick Provisioning

법은 사용하기 전 자원을 미리 할당해 놓기 때문에 데이터의 저장을 위해 자원이 필요할 때 지연 없는 빠른 처리가 가능하다. 또한, 사용자의 데이터가 증가할 경우 이를 지원하기 위한 자원 관리가 직관적이고 쉽다는 장점이 있다. 반면 물리적 자원을 미리 할당해 놓는 방법이기 때문에 사용자의 실제 사용 데이터가 적으면 사용되지 않는 자원이 발생하여 낮은 운영 효율과 실제 데이터 관리 비용이 높은 단점이 있다.

2.2 씬프로비저닝 (Thin Provisioning)

빅데이터의 한가지 특징은 급속도의 데이터 증가 속도이다. 따라서 저장 시스템은 대용량의 저장 자원을 유지해야 한다. 그러나 예측할 수 없는 데이터의 양을 기반으로 대용량의 저장 시스템을 구축할 경우 실제 사용되는 데이터의 양에 따라 자원의 낭비와 과도한 초기 비용이 발생한다. 이러한 문제를 해결하기 위해 씬프로비저닝 기법이 개발되었다. 이 방법은 사용자에게 보이는 논리적 자원과 비교하여 실제 사용 데이터 용량만큼의 적은 물리적 자원을 할당하는 방법이다[2, 3].

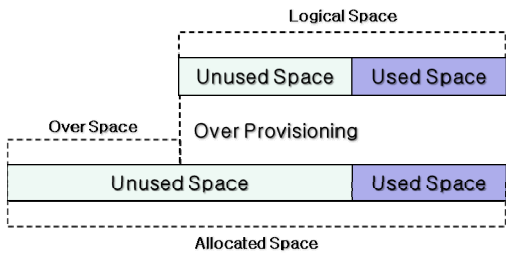


[Fig. 2] Thin Provisioning

Fig. 2는 썬프로비저닝의 기본적인 자원 할당 방법을 보여주고 있다. 그림에서는 사용자에게 보여주는 자원은 데이터를 위해 실제로 사용된 영역뿐만 아니라 사용되지 않은 영역까지 사용자가 요청한 모든 논리적 공간을 보여준다. 그러나 내부적으로는 실제로 사용된 공간의 자원만 물리적으로 할당하였다. 따라서 썬프로비저닝은 저장 시스템의 초기 비용을 줄이고 썬프로비저닝과 비교하여 유연한 자원 관리가 가능하다. 반면 사용하는 데이터가 증가할 경우 추가적인 자원 할당을 관리하기 위한 비용이 발생하며 이 과정에서 필연적으로 발생하는 지연된 작업 처리 속도 때문에 성능이 저하되는 단점이 있다.

2.3 오버프로비저닝 (Over Provisioning)

사용자의 논리적 데이터와 시스템의 물리적 자원을 관리하는 또 다른 방법은 논리적 자원 대비 물리적 자원을 초과로 할당하는 오버프로비저닝 기법이 있다[4-8].

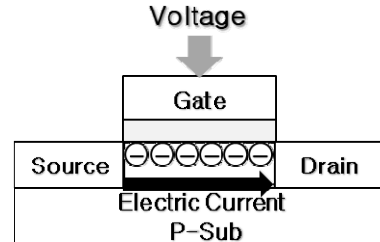


[Fig. 3] Over Provisioning

Fig. 3은 오버프로비저닝을 보여주고 있다. 그림에서 사용자가 요청한 논리적 공간은 사용되지 않은 공간과 실제 데이터를 위한 공간으로 구성되어 있다. 오버프로비저닝은 이 요청을 위해 저장장치에 사용자의 요청보다 더 많은 자원을 할당하였다. 이러한 할당방법은 자원을 추가로 제공해야 하는 비용 문제 때문에 일반적인 저장장치에서는 사용되지 않는다. 그러나 플래시 메모리를 미디어로 사용하는 SSD(solid state disk)에서는 데이터를 쓰기 전 지워져야 하는 특징이 있다. 이 특징은 외부의 쓰기 요청 대비 내부에서 발생하는 쓰기 연산을 표시하는 WAF(write amplification factor)[9-11]을 높이기 때문에 수명을 저하하는 문제가 있다. 그러나 오버프로비저닝을 통해 추가된 초과 공간을 컨트롤러에 할당함으로써 쓰기 및 지우기 연산을 지연시키고 수명 및 내구성을 향상해준다. 따라서 오버프로비저닝은 플래시 기반 저장장치에서 일반적으로 사용되고 있다.

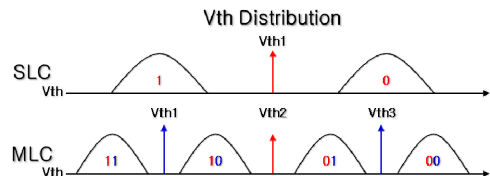
2.4 플래시 메모리 셀의 데이터 구분

플래시 메모리는 전압을 가했을 때 전류 흐름의 여부에 따라 셀의 데이터를 0과 1로 구분한다.



[Fig. 4] Flash Memory Cell

Fig. 4는 플래시 메모리 셀의 동작을 보여주고 있다. 그림과 같이 일정 전압을 가했을 때 플래시 메모리에 쌓여있는 전자의 양에 따라 소스와 드레인 사이에 전자로 이루어진 플로팅 게이트가 생성된다. 그리고 플로팅 게이트를 통해 전류가 흐른다. 그러나 만약 셀 내부에 전자가 부족하면 낮은 전압으로는 전류가 흐를 수 있는 플로팅 게이트가 생성되지 않는다. 이러한 특징을 이용하여 서로 다른 여러 개의 전압을 기준으로 전류가 흐를 수 있도록 구분하면 하나의 셀로 여러 데이터를 표현할 수 있다.



[Fig. 5] Stored Method of Cell

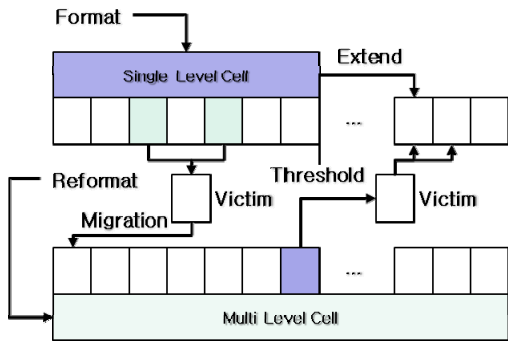
Fig. 5는 SLC(single level cell)[12, 13]와 MLC(multi level cell)[14, 15] 셀의 저장방식을 보여주고 있다. 그림과 같이 SLC는 단일 전압인 V_{th1} 을 기준으로 전류의 흐름 여부를 판단하기 때문에 1과 0의 데이터를 식별한다. 그리고 MLC는 전압을 V_{th1} 에서 V_{th3} 까지 단계적으로 변경하며 전류가 마지막으로 흐르는 구간을 식별하여 셀이 유지하고 있는 데이터를 식별한다. 예를 들어 전압 레벨 V_{th1} 에서는 전류가 흐르지만 V_{th2} 에서 전류가 흐르지 않는 경우 셀이 유지하고 있는 데이터는 10이다. 이러한 방법으로 다중 비트(multi bits)를 유지하는 메모리 관리 방법은 동일한 용량의 플래시 메모리에서 유지할 수 있는 데이터를 증가시킬 수 있는 장점이 있다. 그

러나 다중 비트를 유지할 경우 여러 개의 기준 전압으로 구분할 수 있는 셀들의 특성 간격이 좁아서 수명이 줄어드는 단점이 있다. 따라서 일반적인 SSD에서는 사용 빈도가 많은 메타데이터는 SLC 셀에서 관리하고 일반 데이터는 다중 비트 셀에서 관리한다.

3. 자원 최적화 기법

3.1 핵심 아이디어

본 논문에서 제안하는 기법의 핵심 아이디어는 플래시 메모리를 단일 비트 셀과 다중 비트 셀로 포맷하고 데이터의 사용 빈도에 따라 다른 포맷의 메모리로 데이터를 관리하는 방법이다.

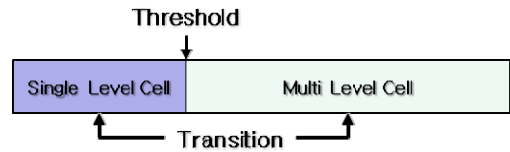


[Fig. 6] Key Idea

Fig. 6은 논문에서 제안하는 자원 최적화 기법의 핵심 아이디어를 보여주고 있다. 자원 최적화 기법은 Fig. 6과 같이 플래시 메모리 블록을 임계치 구간까지는 단일 레벨 셀로 포맷하여 사용된다. 그리고 임계치에 도달했을 때 사용량이 많지 않은 데이터를 모아 멀티 레벨 셀로 포맷된 블록으로 이동시킨다. 그리고 주기적으로 멀티 레벨 셀에 저장된 데이터 중 사용 빈도가 높은 데이터는 단일 레벨 셀의 블록으로 이동시킨다. 마지막으로 전체 데이터 사용량과 사용 빈도에 따라 단일 레벨 셀 블록과 멀티 레벨 셀 블록의 임계치를 조절하여 플래시 메모리 사용량을 최적화한다.

3.2 포맷 정책

Fig. 7은 포맷 정책을 보여주고 있다. 그림과 플래시 메모리는 단일 레벨 셀과 멀티 레벨 셀로 하이브리드 포맷되어 관리되며 이 경계는 저장장치에 사용하는 데이터

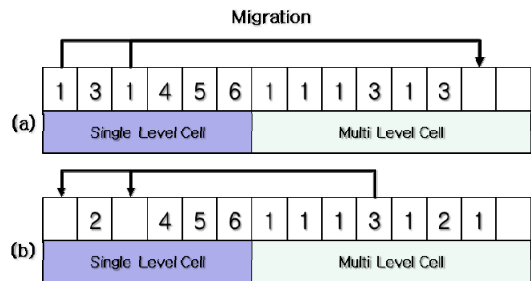


[Fig. 7] Format Policy

의 패턴에 따라 임의의 비율로 정해진다. 데이터의 패턴은 블록별 사용되는 횟수를 기반으로 정해지며, 처음 정해진 경계값은 데이터의 사용량과 사용 빈도에 따라 조절된다. 예를 들어 실제 사용량이 전체 저장용량 자원을 50% 이하로 사용할 경우 가용한 멀티 레벨 셀 블록을 단일 레벨 셀 블록으로 전환한다. 반면 자원 사용률이 90% 이상인 경우 단일 레벨 셀 블록에서 사용빈도가 낮은 데이터를 멀티레벨 셀 블록으로 이전하고 해당 블록은 멀티레벨 셀 블록으로 전환한다.

3.3 데이터 이전 정책

Fig. 8은 데이터 이전 정책을 보여주고 있다. Fig. 8의 (a)에서는 단일 레벨 셀 블록이 가득 찬 이후 블록을 선별하여 멀티 레벨 셀 블록으로 이전하는 방법을 보여주고 있다. 블록을 선별하는 방법은 임의의 크기의 데이터 쓰기가 완료되었을 때 사용된 횟수를 기반으로 사용횟수가 적은 블록을 선정한다. 그림의 예제에서는 첫 번째와 세 번째 블록의 사용횟수가 1로 가장 적은 횟수의 사용을 하였다. 따라서 이 블록들이 멀티 레벨 셀 블록으로 이전하기 위한 블록으로 선정되었다. 그리고 Fig. 8의 (b)는 멀티 레벨 셀 블록에서 단일 레벨 셀 블록으로 이전하는 방법을 보여주고 있다. 이 방법도 블록의 사용횟수를 기반으로 이전 블록을 선정한다. 이전 블록을 선정하는 주기는 단일 레벨 셀 블록으로부터 블록들이 이전해 온 이후나 특정 횟수의 읽기 쓰기 동작을 수행할 때마다 주기적으로 전환한다. 그림의 예제에서는 멀티 레벨 셀 블록

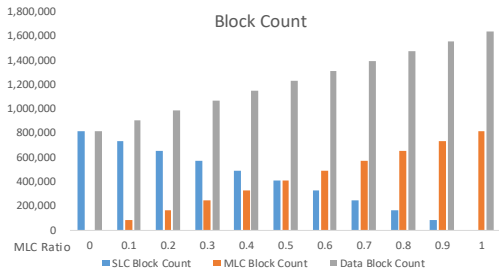


[Fig. 8] Migration Policy

중 네 번째 블록이 사용횟수가 3이고 다른 블록들보다 사용 빈도가 높다. 따라서 이 블록이 단일 레벨 셀 블록으로 이전하기 위한 블록으로 선정되었다.

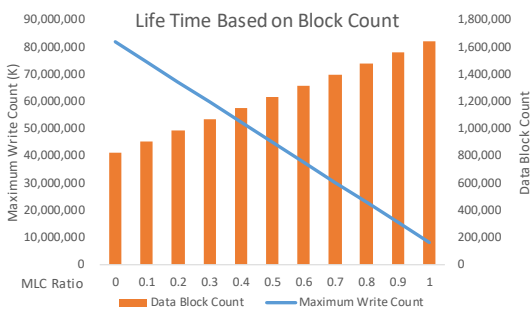
4. 실험 및 평가

본 논문에서는 시뮬레이션을 통해 자원 최적화 기법의 장점을 분석하였다. 시뮬레이션한 SSD의 용량은 100GB이다. 페이지의 크기는 1KB이고, 각 블록의 페이지 수는 128개다. 또한, SLC와 MLC의 수명은 각각 10만 번과 1만 번 반복 쓰기 가능하다고 가정하였다.



[Fig. 9] Block Count

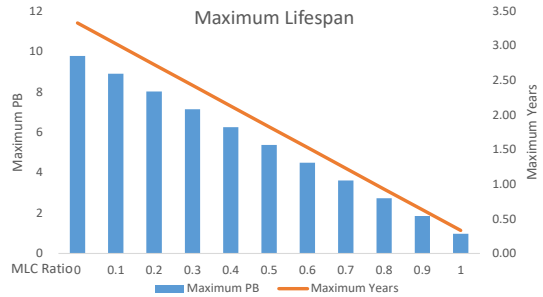
Fig. 9는 SLC와 MLC 블록의 비율에 따라 변화하는 블록의 숫자를 보여주고 있다. 그림에서 X축은 MLC 블록의 비율을 의미하고 Y축은 블록의 수이다. 물리적 블록의 수는 819,200개이나 MLC 비율에 따라 데이터를 저장할 수 있는 데이터 블록의 자원은 최대 1,638,400개까지 100% 증가하였다.



[Fig. 10] Life Time

Fig. 10은 MLC 블록 비율에 따른 블록의 수와 최대 쓰기 횟수를 보여주고 있다. 실험에서 수명은 데이터 블록의 증가와 관계없이 81,920,000K 회에서 8,192,000K

회로 SLC 블록 수에 따라 수명이 최대 10배 차이가 발생했다. 이러한 원인은 SLC와 MLC의 반복 쓰기 가능한 수명은 10배의 차이가 있기 때문이다.



[Fig. 11] Maximum Lifespan

Fig. 11은 MLC 블록 비율에 따른 최대 수명을 보여주고 있다. 실험에서 SSD의 성능은 100MB/s로 가정하였다. 이 성능을 1일로 환산하면 하루 최대 8.64TB의 데이터를 처리할 수 있다. 따라서 MLC 블록 비율에 따라 각 SSD는 1PB에서 10PB 처리할 수 있고, 수명은 최대 7개월에서 3년 4개월이다.

5. 결론

본 논문에서는 핫 데이터와 콜드 데이터를 구분하여 SLC와 MLC 타입 블록을 관리하기 위한 하이브리드 포맷 기법을 제안하였다. 제안한 기법은 데이터의 필요한 용량과 데이터 사용 빈도에 따라 저장장치의 용량과 수명을 조절할 수 있다. 또한 실험을 통하여 MLC 블록의 비율로 조절할 수 있는 용량과 수명의 증감 효과를 측정하였다. 그러나 멀티 레벨 블록의 타입과 실시간 변화하는 데이터 타입을 프로비저닝 기법에 반영하는 본질적인 연구가 필요하다. 향후에는 핫데이터를 실시간 분석하여 적응형으로 포맷하는 방법을 연구할 예정이다.

REFERENCES

- [1] H.Fenton and P.Kennedy, "ESXi on Arm on Pi and the Post-virtualized Datacenter," Running ESXi on a Raspberry Pi, pp.17-45, 2021.
- [2] L.Shrestha and N.J.Sheikh, "Multiperspective Assessment of Enterprise Data Storage Systems: Literature Review,"

2022 Portland International Conference on Management of Engineering and Technology, pp.1-8, 2022.

- [3] Y.Zou, A.Awad and M.Lin, "DirectNVM: Hardware-accelerated NVMe SSDs for High-performance Embedded Computing," ACM Transactions on Embedded Computing Systems, Vol.21, No.9, pp.1-24, 2022.
- [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] M.Kumar, S.C.Sharma, A.Goel and S.P.Singh, "A comprehensive survey for scheduling techniques in cloud computing," Journal of Network and Computer Applications, Vol.143, No.1, pp.1-33, 2019.
- [6] W.M.Danquah and D.T.Altılar, "Vehicular Cloud Resource Management, Issues and Challenges: A Survey," IEEE Access, vol.8, pp.180587-180607, 2020.
- [7] R.Liu, Z.Tan, L.Long, Y.Wu, Y.Tan and D.Liu, "Improving Fairness for SSD Devices through DRAM Over-Provisioning Cache Management," IEEE Transactions on Parallel and Distributed Systems, vol.33, no.10, pp.2444-2454, 2022.
- [8] N.Y.Ahn and D.H.Lee, "Forensic Issues and Techniques to Improve Security in SSD With Flex Capacity Feature," IEEE Access, vol.9, pp.167067-167075, 2021.
- [9] J.Yang, S.Pei and Q.Yang, "WARCIP: Write Amplification Reduction by Clustering I/O Pages," 12th ACM International Conference on Systems and Storage, pp.155-166, 2019.
- [10] A.Eisenman, A.Cidon, E.Pergament, R.Stutsman, M.Alizadeh and S.Katti, Flashield: a Hybrid Key-value Cache that Controls Flash Write Amplification," NSDI '19 Open Access Sponsored by NetApp, 2020.
- [11] W.Zhang, X.Zhao, S.Jiang and H.Jiang, "ChameleonDB: a Key-Value Store for Optane Persistent Memory," Sixteenth European Conference on Computer Systems, pp.194-209, 2021.
- [12] L.Kong, Y.Liu, H.Liu and S.Zhao, "Protograph QC-LDPC and Rate-Adaptive Polar Codes Design for MLC NAND Flash Memories," IEEE Access, Vol.7, pp.37131-37140, 2019
- [13] G.Hemink and A.Goda, "5 - NAND Flash Technology Status and Perspectives," Semiconductor Memories and Systems, pp.119-158, 2022
- [14] L.Kong, Y.Liu, H.Liu and S.Zhao, "Protograph QC-LDPC and Rate-Adaptive Polar Codes Design for MLC NAND Flash Memories," IEEE Access, Vol.7, pp.37131-37140, 2019
- [15] R.He,H.Hu,C.Xiong and G.Han, "Artificial Neural Network Assisted Error Correction for MLC NAND Flash Memory," micromachines, Vol.12, No.8 pp.879, 2021.

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