

낸드 플래시 메모리 기반 저장 장치에서 다양한 초과 제공을 통한 성능 분석 및 예측

이현섭

백석대학교 컴퓨터공학부 조교수

Performance analysis and prediction through various over-provision on NAND flash memory based storage

Hyun-Seob Lee

Assistant Professor, Division of Computer Engineering, Baekseok University

요 약 최근 급격한 기술의 발달로 다양한 시스템에서 발생하는 데이터양이 증가하고 있으며, 많은 양의 빅데이터(big data)를 처리해야 하는 엔터프라이즈 서버(enterprise server)와 데이터 센터(data center)의 경우 비용이 증가하더라도 높은 안정성과 고성능의 저장 장치를 적용하는 것이 필요하다. 이러한 시스템에서는 고성능의 읽기/쓰기 성능을 제공하는 SSD(solid state disk)를 저장 장치로 사용하는 경우가 많다. 그러나, 페이지 단위로 읽기 쓰기를 하고 블록 단위로 지우기 연산을 해야하고 쓰기 전 지우기 연산을 수행해야 하는 특징 때문에 중복 쓰기가 다발할 경우 성능이 저하되는 문제가 있다. 따라서 이러한 성능 저하 문제를 지연시키기 위해 SSD의 내부적으로 초과 제공(over-provision) 기술을 적용하고 있다. 그러나 초과 제공 기술은 성능 대신 많은 저장공간의 비용을 소모하는 단점이 있기 때문에 적정 성능 이상의 비효율적인 기술의 적용은 과도한 비용을 지불하게 만드는 문제가 있다. 본 논문에서는 SSD에서 다양한 초과 제공을 적용하였을 때 발생하는 성능과 비용을 측정하고, 이를 기반으로 시스템에 최적화된 초과 제공 비율을 예측하는 방법을 제안했다. 본 연구를 통해 빅데이터를 처리하는 시스템에서 성능의 요구사항을 만족하기 위한 비용과의 절충점(trade-off)를 찾을 수 있을 것으로 기대한다.

주제어 : 낸드 플래시 메모리, 초과 제공, 성능최적화, 저장시스템, 빅데이터

Abstract Recently, With the recent rapid development of technology, the amount of data generated by various systems is increasing, and enterprise servers and data centers that have to handle large amounts of big data need to apply high-stability and high-performance storage devices even if costs increase. In such systems, SSD(solid state disk) that provide high performance of read/write are often used as storage devices. However, due to the characteristics of reading and writing on a page-by-page basis, erasing operations on a block basis, and erasing-before-writing, there is a problem that performance is degraded when duplicate writes occur. Therefore, in order to delay this performance degradation problem, over-provision technology of SSD has been applied internally. However, since over-provided technologies have the disadvantage of consuming a lot of storage space instead of performance, the application of inefficient technologies above the right performance has a problem of over-costing. In this paper, we proposed a method of measuring the performance and cost incurred when various over-provisions are applied in an SSD and predicting the system-optimized over-provided ratio based on this. Through this research, we expect to find a trade-off with costs to meet the performance requirements in systems that process big data.

Key Words : NAND flash memory, Overprovision, Performance Optimization, Storage System, Big Data.

*This paper was supported by 2022 Baekseok University Research Fund

*Corresponding Author : Hyun-Seob Lee(hyunseob@bu.ac.kr)

Received February 26, 2022

Revised March 15, 2022

Accepted March 20, 2022

Published March 28, 2022

1. 서론

낸드 플래시 메모리는 저전력 고성능의 장점이 있기 때문에 다양한 저장 장치의 미디어로 활용되고 있다. 특히 실시간으로 많은 양의 빅데이터를 처리해야 하는 엔터프라이즈 서버와 데이터 센터의 높은 성능 요구를 만족해야 하기 때문에 보편적으로 플래시 메모리로 구성된 SSD(solid state disk)가 2차 저장 장치로 활용되고 있다. 그러나 플래시 메모리는 데이터를 쓰기 전 해당 공간에 지우기 연산이 선수행 되어 있어야 하며, 읽기 쓰기 연산은 페이지 단위로 처리되는 반면 지우기 연산은 블록 단위로 수행되는 비대칭 연산의 독특한 특징이 있다.[1-4] 이러한 특성을 감추기 위해 논리적인 주소를 물리적인 주소로 매핑하는 FTL(Flash Translation Layer)이 사용되고 있으나 빈번한 덮어쓰기(overwrite)가 발생할 경우 성능이 저하되는 것은 막을 수 없다.[5-8] 이러한 성능 저하 문제를 가리기 위해 사용되는 한 가지 방법은 초과 제공이 있다.[9-15] 이 방법은 논리적인 저장 장치의 크기보다 더 많은 물리적인 저장 장치를 제공하여 중복 쓰기로 인해 발생하는 쓰기 지우기 횟수를 지연시키는 데 도움을 준다. 그러나 초과 제공은 물리적인 공간을 추가적으로 제공해야 하는 비용의 문제가 있기 때문에 요구 성능을 만족하기 위한 최적화된 초과 제공을 사용하는 것이 필요하다. 본 논문에서는 실험을 통하여 단계별 초과 제공의 비용과 성능을 측정하고 필요 성능에 최적화된 초과 제공의 비용을 설정하는 방법을 제안한다.

2. 플래시 메모리 기반 저장 장치의 특징

2.1 플래시 메모리의 구성

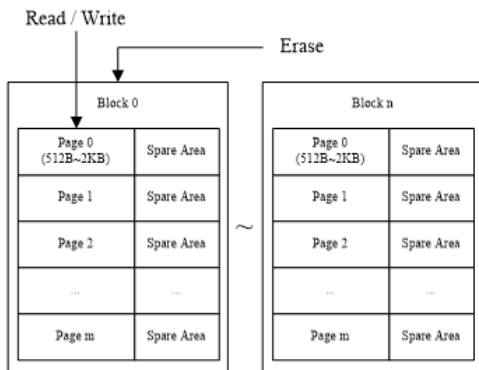


Fig. 1. Composition of Flash Memory

Fig.1은 플래시 메모리의 구성을 보여주고 있다. 그림과 같이 플래시 메모리는 n 개의 블록으로 구성되어 있고 m 개의 페이지로 구성되어 있다. 각 페이지의 용량은 512B에서 2KB까지 낸드 단품의 구성에 따라 다르고 블록을 구성하는 페이지의 개수인 m 또한 32개에서 256개 등 다양한 제품군이 존재한다. 그리고 각 페이지는 페이지에 대한 메타정보를 유지하기 위한 스페어 공간이 있고 이 공간에는 ECC(error collection code) 등 각 페이지에 대한 다양한 정보를 포함하고 있다.

2.2 플래시 메모리의 특징

낸드 플래시 메모리는 사용할 때 저 전력 소비의 특징과 빠른 데이터 읽고 쓰기 속도의 장점이 있다. 이러한 장점 때문에 소규모 시스템에서 빅데이터를 관리하는 대용량 엔터프라이즈 서버까지 다양한 분야의 저장 장치 미디어로 활용되고 있다. 그러나 플래시 메모리는 비대칭의 읽고 쓰는 성능을 가지고 있고, 이미 데이터가 기록된 공간에 데이터를 쓰기 위해서는 이전의 데이터를 지워야 하는 특징과 데이터를 쓰는 단위와 지우는 단위가 다르다는 특징이 있다. 따라서 덮어쓰기가 발생하면 블록 내에 있는 다른 페이지의 데이터를 비어있는(지워져 있는) 블록으로 이동하고 덮어쓰기 데이터를 다시 쓰는 연산을 수행해야 한다. 이러한 독특한 특징을 감추고 데이터를 처리하기 위해 FTL이라는 소프트웨어 계층을 사용하고 있다.

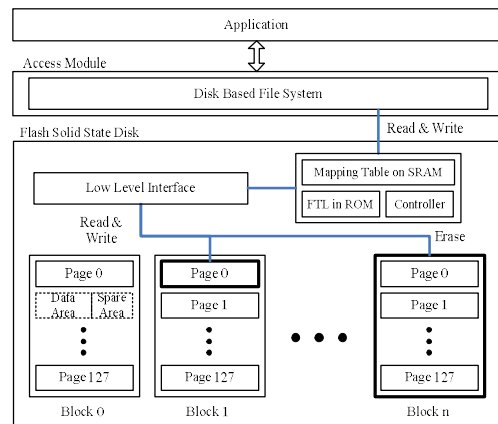
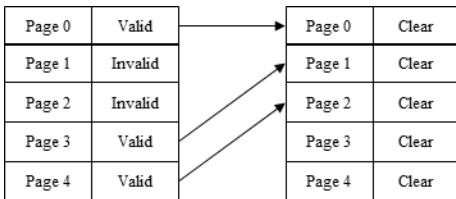


Fig. 2. Architecture of Solid State Disk

Fig.2는 플래시 메모리의 특성을 감추기 위해 FTL을 적용한 SSD의 예제를 보여주고 있다. FTL은 논리적인 주소를 물리적인 주소로 매핑(mapping) 하는 기능을

제공하여 덮어쓰기가 발생하여도 많은 양의 데이터 이동과 지우기 연산을 수반해야 하는 연산들을 지연시킬 수 있다. 예를 들어 논리적인 주소 10번의 데이터가 물리적인 주소 20번의 페이지에 매핑되어 있을 때 논리적인 주소 10번을 주소로 하는 데이터에 대한 덮어쓰기가 발생했을 경우 이전 물리적인 20번의 페이지는 무효화(invalidate) 하고, 비어있는 새로운 물리적인 페이지 21번 페이지를 할당받아서 데이터를 쓴다. 그리고 논리적인 주소 10번의 데이터는 물리적인 페이지 21번에 매핑되어 있다는 정보를 갱신하는 방식으로 데이터를 관리하는 방법이다. 따라서 FTL을 적용한 SSD의 성능은 덮어쓰기가 발생했을 때 매핑을 최적화 할수록 성능이 향상되는 특징이 있다.



- Step:
1. Copy all valid data to the new block (free block)
 2. Make a free block with erasing the old block

Fig. 3. Copy-back Operation

Fig. 3는 데이터를 저장할 수 있는 페이지가 부족할 때 지우기 연산을 통해 데이터를 모두 지운 빈 블록(free block)을 만드는 카피 백(copy back) 연산을 보여주고 있다. 이 연산은 선택된 이전 블록(old block)으로부터 유효한(valid) 데이터를 읽어서 빈 블록에 다시 쓴 다음 이전 블록은 지우기 연산을 통해 새로운 빈 블록을 만드는 방법이다. 이 연산을 통해 비어있는 페이지를 추가로 만들 수 있다. 그러나 이 과정에서 많은 읽기와 쓰기 연산이 발생하여 성능이 저하되는 문제가 있다. 이러한 문제를 줄이기 위해서는 카피 백 연산을 수행하기 위한 블록을 선택할 때 무효(Invalid) 데이터가 많은 블록을 선택하는 알고리즘이 필요하다. 무효 페이지가 많은 경우 카피 백 연산을 수행할 때 읽기와 쓰기 연산이 감소하기 때문에 성능저하가 감소하는 효과가 있다.

3. 초과 제공 기법

3.1 초과 제공 기법의 구조

저장 장치의 내부적인 공간 제공 기술은 크게 얇은 제공(thin provision)과 초과 제공(over provision)으로 구분된다. 본 연구에서는 초과 제공 기술에 대해 설명하고 낸드 플래시 메모리로 구성된 SSD의 성능 향상에 초과 제공 기법이 적합한 기술임을 설명한다.

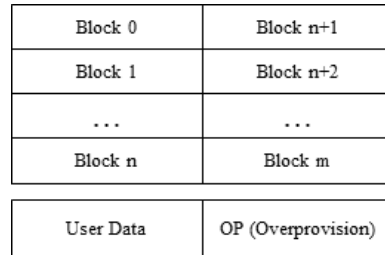


Fig. 4. Design of Overprovision

Fig. 4는 초과 제공 기법의 설계 구조를 보여주고 있다. 그림과 같이 용량을 제공하기 위한 일반적인 사용자 데이터(user data) 공간은 n 개의 블록으로 구성되어 있고 OP(초과 제공의 비율)에 따라 m 개의 추가 블록이 제공된다. 외부에서 확인할 수 있는 저장 장치의 블록의 개수는 용량에 해당하는 n 개의 블록이지만 내부적으로 데이터를 수용하는 블록의 개수는 OP에 따라 m 개의 블록으로 구성된다.

3.2 SSD에서 초과 제공의 장점

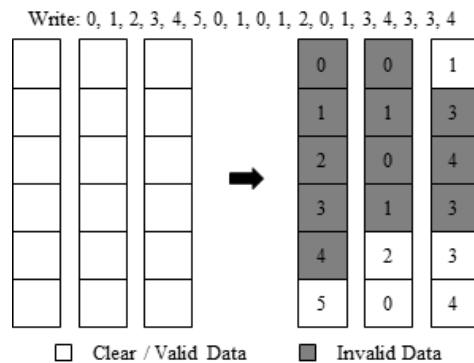


Fig. 5. Example of No Overprovision

SSD는 빈 블록이 부족할 경우 성능이 저하되어도 카피 백 연산을 통해 데이터 공간을 확보해야 한다. 이때 카피 백 연산을 위해 선택된 블록 내에 무효화 페이지가 많을수록 읽기/쓰기 연산이 감소하기 때문에 이러한 블록이 많을수록 성능 저하를 줄일 수 있다. OP는 초과 제

공된 블록을 통해 무효화 페이지가 많아지게 만드는 효과가 있다.

Fig. 5는 OP를 적용하지 않은 SSD에서 쓰기 연산을 수행한 예제를 보여주고 있다. 그림의 예제에서 플래시 메모리는 3개의 블록으로 구성되어 있고 각 블록은 6개의 페이지로 구성되어 있다고 가정하였다. 그리고 18번의 쓰기 연산이 수행되었을 때 유효/무효 데이터를 유지하고 있는 상태를 보여주고 있다. 이 구성에서는 18번의 쓰기 연산 이후 카피 백 연산을 수행해야 하며 각 블록은 평균 2개의 유효 데이터 페이지와 4개의 무효 데이터 페이지로 구성된다. 그리고 카피 백 연산을 수행하면 1번의 읽기와 쓰기 연산이 수행된다.

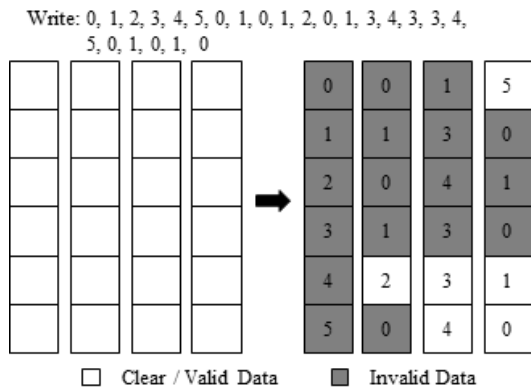


Fig. 6. Example of Overprovision

Fig.6은 OP를 적용한 SSD에서 쓰기 연산을 수행한 예제를 보여주고 있다. 그림의 예제에서 플래시 메모리는 3개의 블록과 1개의 OP 블록으로 구성되어 있다고 가정하였다. 그리고 이 구성에서는 24번의 쓰기 연산을 수행할 때까지 카피 백 연산을 지연시킬 수 있다. 이 예제를 OP가 없는 예제와 비교하면 각 블록은 평균 1.5개의 유효 데이터 페이지와 4.5개의 무효 데이터 페이지로 구성되며 카피 백 연산을 수행했을 때 추가적인 읽기 쓰기 연산 없이 지우기 연산을 수행하는 것 만으로 빈 블록을 생성할 수 있는 장점이 있다.

이러한 예제의 비교를 통해 OP 블록을 사용하는 경우가 OP 블록을 사용하지 않는 경우보다 평균 유효 데이터 페이지 개수가 적고 평균 유효 데이터 페이지가 많기 때문에 카피 백 연산을 수행할 때 성능 저하를 줄일 수 있는 장점이 있다는 것을 알 수 있다. 그러나 OP 블록을 추가하는 것은 하드웨어적인 비용이 발생하기 때문에 성

능 향상을 위해 OP를 적용하는 것은 성능과 비용 사이의 절충점 (trade-off)이고 최적화된 절충점을 찾는 것은 어려운 문제이다. 따라서 본 논문에서는 다양한 초과 제공 실험을 통해 OP 별 성능을 예측하고 성능과 비용 사이의 적절한 절충점을 찾기 위한 방법을 분석한다.

4. 실험 및 결과

본 논문에서는 100MB의 SSD 환경에서 페이지 단위 맵핑을 하는 FTL을 이용하여 데이터의 쓰기 연산을 수행하는 시뮬레이션을 수행하였다. 실험에서 사용한 데이터는 100MB 용량 범위 주소를 갖는 데이터를 임의의 순서로 중복 되도록 발생시켰고 각 실험마다 처리된 쓰기 데이터량은 100GB이다. 시뮬레이션에서 가정한 설정은 각 블록은 128개의 페이지로 구성되어 있다고 가정하였고 각 페이지의 용량은 1KB로 가정하였다. 또한 사용된 기본 사용자 블록의 개수는 800개로 정하였다. 그리고 실험에서는 용량의 10%씩 초과 제공을 증가하는 방향으로 설정하였다. 마지막으로 읽기/쓰기/지우기 연산의 시간적 비용은 1:10:100으로 가정하였고 읽기 연산의 속도는 10us로 가정하였다.

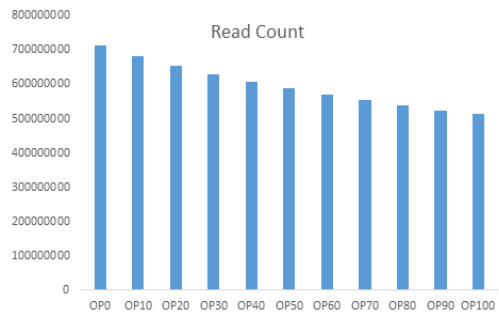


Fig. 7. Result of Read Count

Fig.7은 읽기 성능의 결과를 보여주고 있다. 그림에서 X 축은 OP를 의미하고 Y 축은 읽기 횟수를 의미한다. OP는 0%에서 100%까지 10%씩 증가하며 실험을 했고 읽기 횟수는 OP가 증가할수록 좋은 성능을 보이는 경향을 보였고, OP가 0%일 때 713,694,272회 OP가 100%일 때 512,956,140회 발생했다. OP가 증가할 때 OP 0%와 비교하여 성능이 개선된 비율은 4.27%에서 38.27%까지 개선된 것을 확인할 수 있었다.

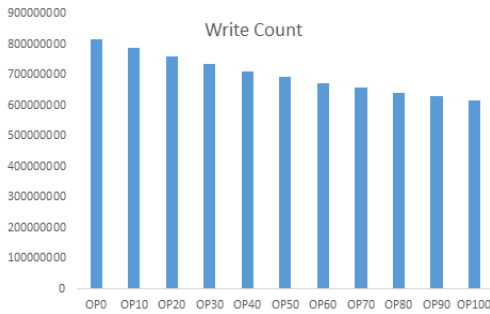


Fig. 8. Result of Write Count

Fig.8은 쓰기 성능의 결과를 보여주고 있다. 쓰기 횟수는 OP가 증가할수록 좋은 성능을 보이는 경향을 보였고, OP가 0%일 때 818,551,872회 OP가 100%일 때 617,813,740회 발생했다. OP가 증가할 때 OP 0%와 비교하여 개선된 성능의 비율은 3.72%에서 31.89% 개선된 것을 확인할 수 있었다.

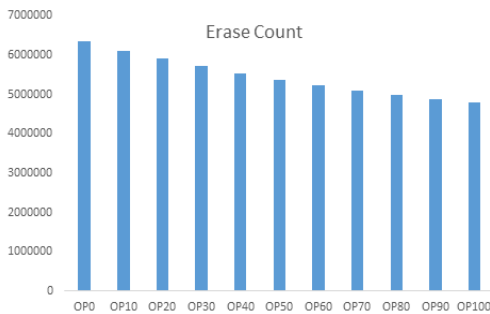


Fig. 9. Result of Erase Count

Fig.9는 지우기 성능의 결과를 보여주고 있다. 전체적으로 지우기 횟수는 OP가 증가할수록 좋은 성능을 보이는 경향을 보였다. 이것은 카피 백 연산이 OP가 증가할수록 줄어드는 것을 의미한다. 실험의 결과에서 지우기 횟수는 OP가 0%일 때 6,357,972회 OP가 100%일 때 4,794,936회 발생했다. OP가 증가할 때 OP 0%와 비교하여 개선된 성능의 비율은 3.73%에서 32% 개선된 것을 확인할 수 있었다.

마지막으로 Fig.10은 쓰기 연산을 처리하는 동안 소모된 시간의 결과를 보여주고 있다. 실험의 결과에서 소비된 시간은 OP가 0%일 때 약 26시간 12분 38초 OP가 100%일 때 약 19시간 55분 5초 소비되었다. 전체적으로 OP 0%와 비교하여 개선된 성능의 비율은 3.76%에서 32.36% 개선된 것을 확인할 수 있었다.

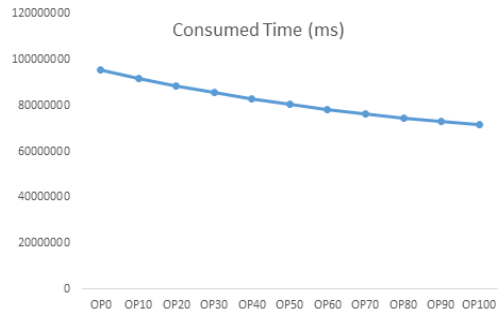


Fig. 10. Result of Consumed Time

5. 결론

본 논문에서는 초과 제공의 비율을 조절한 다양한 OP 환경에서 대용량의 데이터의 쓰기 연산을 수행하여 성능을 측정하였다. 실험에서는 초과 제공을 적용하지 않은 저장 장치와 비교하여 초과 제공을 제공하는 저장 장치의 성능이 3.76 ~ 32.36% 성능이 향상되었고 초과 제공의 비율이 증가할수록 성능이 향상되는 결과를 확인하였다. 본 논문에서는 비록 초과 제공 기법이 물리적인 비용을 추가해야 하는 문제가 있지만 실험 결과를 통해 목표 성능과 비용 사이에서 절충점을 찾을 수 있음을 확인하였다. 향후에는 다양한 성능 파라미터와 함께 능동적으로 초과 제공의 비율을 조절하여 최적화된 성능 개선 방법을 연구할 예정이다.

REFERENCES

- [1] S. S. Chae, R. Mativenga, J. Y. Paik, M. Attique & T. S. Chung. (2020). DSFTL: An efficient FTL for flash memory based storage systems. *Electronics* 9(1), 145.
- [2] W. Xie, Y. Chen & P. C. Poth. (2017). ASA-FTL: An adaptive separation aware flash translation layer for solid state drives. *Parallel Computing*, 61, 3-17.
- [3] I. B. Zion. (2020). Key-value FTL over open channel SSD. *12th ACM International Conference on Systems and Storage*. 192-192.
- [4] H. S. Lee, H. S. Yun & D. H. Lee (2011). HFTL: hybrid flash translation layer based on hot data identification for flash memory. *IEEE Transactions on Consumer Electronics*, (4), 2005-2011.
- [5] H. S. Lee & D. H. Lee. (2010). An efficient index buffer management scheme for implementing a B-tree on NAND flash memory. *Data & Knowledge Engineering*. 69(9), 901-916.

- [6] H. S. Lee, S. W. Park & D. H. Lee (2013). RMSS: an efficient recovery management scheme on NAND flash memory based solid state disk. *IEEE Transactions on Consumer Electronics*, 59(1), 107–112.
- [7] S. Kim & Y. Son. (2021). Optimizing Key–Value Stores for Flash–Based SSDs via Key Reshaping. *IEEE Access* 9, 115135–115144.
- [8] J. H. Park, D. J. Park, T. S. Chung & S. W. Lee. (2021). A Crash Recovery Scheme for a Hybrid Mapping FTL in NAND Flash Storage Devices. *Electronics*, 10(3), 327.
- [9] M. Sacks. (2021). Incentives for the over–provision of public goods. *Journal of Economic Behavior & Organization*, 191, 197–213.
- [10] R. Liu, Z. Tan, L. Long, Y. Wu, Y. Tan & D. Liu. (2022) Improving Fairness for SSD Devices through DRAM Over–Provisioning Cache Management. *IEEE Transactions on Parallel and Distributed Systems*, 1–1.
- [11] Q. Li et al. (2021). RAMBO: Resource Allocation for Microservices Using Bayesian Optimization. *IEEE Computer Architecture Letters*, 20(1), 46–49.
- [12] S. Elashri & A. Azim. (2021). Work–in–Progress: An Energy–Aware Optimization Model for Real–Time Systems Analysis and Design. *International Conference on Embedded Software (EMSOFT)*, 45–46.
- [13] H. Shen & L. Chen. (2018). Resource Demand Misalignment: An Important Factor to Consider for Reducing Resource Over–Provisioning in Cloud Datacenters. *IEEE/ACM Transactions on Networking*, 26(3), 1207–1221.
- [14] L. Chen & H. Shen. (2017). Considering resource demand misalignments to reduce resource over–provisioning in cloud datacenters. *IEEE Conference on Computer Communications*. 1–9.
- [15] D. Bega, M. Gramaglia, M. Fiore, A. Banchs & X. Costa–Pérez. (2020). DeepCog: Optimizing Resource Provisioning in Network Slicing With AI–Based Capacity Forecasting. *IEEE Journal on Selected Areas in Communications*, 38(2), 361–376.

이 현 섭(Hyun–Seob Lee)

[경력]



- 2007년 2월 : 한양대학교 컴퓨터공학과 (공학 석사)
- 2013년 2월 : 한양대학교 컴퓨터공학과 (공학 박사)
- 2012년 3월 ~ 2021년 2월 : 삼성전자 책임연구원
- 2021년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 조교수

퓨터공학부 조교수

· 관심분야 : 인공지능, 저장시스템, 임베디드 시스템

· E–Mail : hyunseob@bu.ac.kr