

<https://doi.org/10.7236/JIIBC.2021.21.2.27>
JIIBC 2021-2-4

모바일 사물인터넷 디바이스를 위한 에너지 효율적인 캐시 및 메모리 관리 기법

Management Technique of Energy-Efficient Cache and Memory for Mobile IoT Devices

반효경*

Hyokyung Bahn*

요약 본 논문은 차세대 사물인터넷 디바이스를 위한 에너지 효율적인 캐시 및 메모리 관리 기법을 제안한다. 제안하는 기법은 전력 소모가 적은 상변화 메모리를 사물인터넷 디바이스의 메인 메모리로 채택하고 캐시 메모리의 관리 시 쓰기 연산에 취약한 상변화 메모리의 쓰기량을 최소화하는 방향으로 설계한다. 구체적으로 살펴보면 최종단 캐시 메모리에서 캐시 블록이 삭제되어 메인 메모리로 반영될 때, 캐시 블록을 구성하는 캐시 라인별 수정 여부를 추적하여 상변화 메모리에 쓰기 발생량을 적게 발생시키는 캐시 블록을 우선적으로 교체한다. 또한, 최종단 캐시 메모리에서 캐시 블록의 참조 비트와 캐시 라인의 수정 비트를 함께 고려함으로써 메모리 시스템의 성능은 훼손하지 않으면서 에너지 소모를 줄이는 방식을 사용한다. 스펙 벤치마크를 이용한 시뮬레이션 실험을 통해 제안한 기법이 상변화 메모리에 발생하는 쓰기량을 평균 34.6% 줄이고 전력 소모를 28.9% 줄이면서 메모리의 성능 저하는 발생시키지 않음을 보인다.

Abstract This paper proposes an energy-efficient cache and memory management scheme for next-generation IoT devices. The proposed scheme adopts a low-power phase-change memory (PCM) as the main memory of IoT devices, aims at minimizing the write traffic to PCM, which is vulnerable to write operations. Specifically, when a cache block of the last-level cache memory is flushed to main memory, the cache block that causes less writes to PCM is preferentially replaced by tracking the modifications of each cache line that constitutes the cache block. In addition, by considering the reference bit of the cache block and the dirty bit of the cache lines, our scheme reduces the energy consumption without degrading the memory system performances. Through simulations using SPEC benchmarks, it is shown that the proposed scheme reduces the write traffic to PCM by 34.6% on average and the power consumption by 28.9%, without memory performance degradations.

Key Words : IoT, last-level cache, phase-change memory, main memory, energy-efficiency

*정회원, 이화여자대학교 컴퓨터공학과
접수일자 2021년 1월 31일, 수정완료 2021년 3월 30일
게재확정일자 2021년 4월 9일

Received: 31 January, 2021 / Revised: 30 March, 2021 /
Accepted: 9 April, 2021
*Corresponding Author: bahn@ewha.ac.kr
Dept. of Computer Engineering, Ewha University, Korea

I. 서 론

최근 차세대 사물인터넷 디바이스의 아키텍처에서 배터리 사용시간 연장을 위해 메인 메모리로 DRAM 대신 전력 소모가 적은 비휘발성 메모리를 사용하는 방안이 각광받고 있다^{1, 2}. 이중 상변화 메모리(phase change memory)는 DRAM에 비해 전력소모가 적고 비휘발성의 특성을 가지고 있어 범용 컴퓨터 시스템의 메인 메모리로 사용하는 방안이 활발히 논의된 바 있다^{3, 4}. 그러나, DRAM에 비해 느린 특성을 가져 범용 컴퓨터 환경에서 DRAM을 대체하기에는 어려운 것으로 평가받고 있다. 이와 달리 사물인터넷 디바이스에서는 연산량이 많지 않고 메모리의 성능보다는 전력 절감 효과가 중요하므로^{5, 6} 상변화 메모리를 메인 메모리로 사용하기에 적절한 환경으로 평가받고 있다^{7, 8}. 한편, 상변화 메모리는 읽기에 비해 쓰기 성능이 더 느린 고유한 특성을 가지고 있다. 특히, 쓰기 지연시간이 DRAM의 7-10배에 이르고 셀별 쓰기 가능 횟수 역시 10^7 - 10^8 번으로 제한적이다³.

이에 따라 본 연구에서는 상변화 메모리가 사물인터넷 디바이스의 메모리로 사용되는 환경을 위한 캐시 및 메모리 관리 기법을 제안한다. 캐시 메모리는 상대적으로 빠른 처리기와 느린 메인 메모리 사이에 존재하며 양자간의 속도 차이를 완충하는 역할을 한다. 그러나, 메인 메모리에 비해 캐시 메모리의 용량이 작기 때문에 캐시 메모리의 공간이 모두 사용 중인 상태에서 새로운 데이터를 캐시 메모리에 적재하기 위해서는 기존에 보관된 데이터를 제거하는 방안이 필요하다. 이때, 캐시 메모리와 메인 메모리 사이에 데이터를 보관하거나 삭제하는 단위로는 캐시 블록을 사용하며, 캐시 메모리에서 삭제할 캐시 블록을 결정하는 알고리즘을 교체 알고리즘이라고 부른다^{9, 10}. 캐시 메모리에서의 교체 알고리즘은 전통적으로 가까운 미래에 사용될 가능성이 낮은 캐시 블록을 교체하여 캐시 적중률을 높이는 것을 목표로 한다. 한편, 캐시 메모리에 존재하는 캐시 블록 중에는 캐시 메모리에 진입한 이후 읽기 연산만 발생한 미수정 블록과 쓰기 연산이 한번이라도 발생한 수정 블록이 있다. 미수정 블록은 캐시 메모리에서 교체될 때 단순히 삭제만 하면 되는 반면 수정 블록은 삭제하기 전에 수정된 내용을 메인 메모리에 반영해야 한다.

본 연구는 수정 블록의 상태를 메인 메모리에 발생시킬 쓰기량에 따라 다단계로 분류하고 캐시 블록의 교체가 필요한 경우 쓰기 발생량이 적은 캐시 블록을 우선적

으로 교체하여 메인 메모리에 발생하는 쓰기량을 줄이는 것을 목표로 한다. 좀 더 구체적으로 설명하면 하나의 캐시 블록을 여러 개의 캐시 라인으로 구분하고 캐시 블록에 쓰기 연산이 발생한 경우 어느 캐시 라인에 발생한 쓰기인지를 캐시 라인별로 표시하여(즉, 라인별로 수정 여부 표시) 캐시 블록의 교체가 필요한 경우 수정된 캐시 라인의 수가 적은 캐시 블록을 우선적으로 교체한다. 이는 메인 메모리로 사용되는 상변화 메모리가 쓰기 연산이 느리고 쓰기 가능 횟수에 제한이 있으므로 가끔씩 쓰기 연산을 최대한 지연시키거나 회피하기 위한 목적이다.

한편, 캐시 메모리의 교체 알고리즘은 캐시 적중률, 즉 처리기가 요청한 데이터 중 캐시 메모리에 존재하여 메인 메모리까지 접근할 필요가 없는 데이터의 비율을 높이는 것이 가장 중요한 목표이다. 따라서, 메인 메모리인 상변화 메모리에 발생하는 쓰기량을 줄이는 것뿐 아니라 캐시 적중률을 떨어뜨리지 않는 것이 중요하다. 이에 본 연구에서는 캐시 메모리의 성능을 떨어뜨리지 않으면서 상변화 메모리에 유발하는 쓰기량을 줄이기 위해 기존의 캐시 교체 알고리즘이 가진 특성, 즉 최근에 사용된 캐시 블록은 삭제하지 않는다는 원칙을 지키면서 수정된 캐시 라인이 가장 적은 캐시 블록을 삭제하는 방안을 제안한다. 제안하는 기법은 스펙 벤치마크를 이용한 시뮬레이션 실험을 통해 캐시 적중률을 훼손하지 않으면서 상변화 메모리에 발생하는 쓰기량을 크게 줄여, 궁극적으로 모바일 사물인터넷 기기의 배터리 수명을 연장할 수 있음을 보인다.

II. 에너지 효율적 캐시 및 메모리 관리

본 장에서는 차세대 사물인터넷 디바이스를 위한 캐시 및 메모리 관리 기법을 상세히 기술한다. 제안하는 기법에서는 기존 아키텍처와 마찬가지로 캐시 메모리로 SRAM을 사용하며, 메인 메모리로는 DRAM 대신 상변화 메모리를 사용한다. 그림 1은 제안하는 관리 기법을 위한 아키텍처를 보여주고 있다. 상변화 메모리로 구성되는 메인 메모리의 바로 상단인 최종단 캐시 메모리의 모습과 그 상단의 중앙처리기를 보여주고 있다. 중앙처리기 내에는 1단 및 2단 캐시 메모리가 존재하며, 최종단 캐시 메모리는 그 윗단의 2단 캐시 메모리와 캐시 라인 단위로 데이터를 주고 받는 반면, 아랫단의 메인 메모리와는 캐시 블록 단위로 데이터를 주고 받는다. 제안하는 기법에서 최종단 캐시메모리의 운영은 n -방향 연관 캐

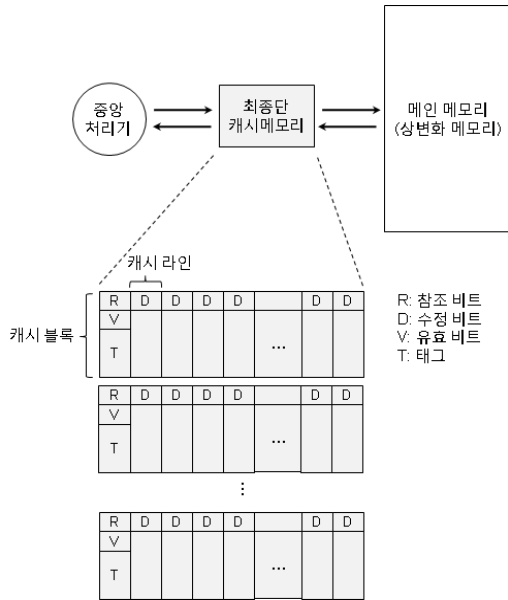
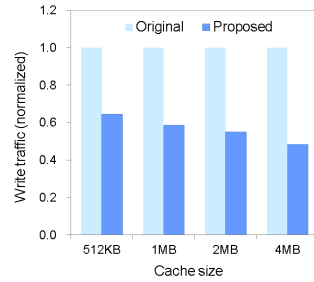


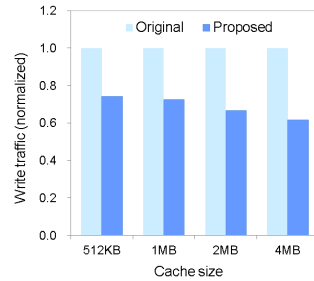
그림 1. 제안 기법의 캐시 메모리 아키텍처.
 Fig. 1. Cache memory architecture of the proposed scheme.

시, 즉 각 캐시 블록이 진입할 수 있는 캐시 메모리 상의 위치가 서로 다른 n 군데인 캐시 메모리 운영 방법을 사용한다. 그림에서 캐시 블록 하나의 크기는 4KB이고 캐시 라인의 크기는 256B로, 캐시 블록 하나가 16개의 캐시 라인으로 구성된다.

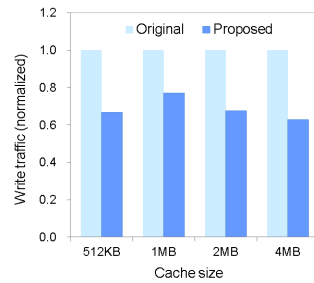
각각의 캐시 블록에는 참조 비트(reference bit)와 유효 비트(valid bit)를 두어 최근에 참조가 일어났는지와 유효한 데이터를 보관하고 있는지를 표시한다. 캐시 블록이 캐시 메모리에 진입한 이후 수정된 부분이 있는지를 표시하기 위해 각 캐시 블록은 수정 비트(dirty bit)를 유지한다. 수정 비트는 해당 캐시 블록이 캐시 메모리에서 교체될 때 메인 메모리에 반영해야 하는지 여부를 표시한다. 본 연구에서 제안하는 기법은 캐시 블록이 아닌 캐시 라인별로 수정 비트를 두어 해당 캐시 블록이 캐시 메모리에서 삭제될 때 발생할 쓰기량을 미리 예측한다. 즉, 16개의 캐시 라인별 수정 비트를 보고 해당 캐시 블록이 교체될 때 몇 개의 캐시 라인을 메인 메모리에 반영해야 하는지를 체크한다. 수정 비트가 1인 캐시 라인은 해당 라인의 내용이 캐시 메모리에 진입한 이후 적어도 1번 이상의 쓰기 연산이 발생한 것을 의미하므로 이를 포함한 캐시 블록이 교체될 때 메인 메모리에 반영되어야 한다.



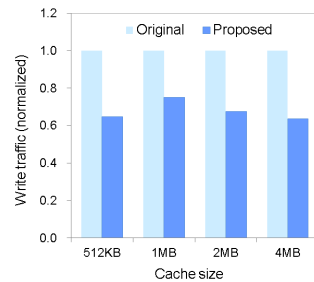
(a) bwaves



(b) bzip2



(c) gcc



(d) sieng

그림 2. 제안 기법의 상변화 메모리에 대한 쓰기 절감 효과
 Fig. 2. Reduced write traffic by adopting the proposed scheme.

본 연구에서 제안하는 기법은 각 캐시 블록의 참조 비트와 수정 비트들의 상태를 보고 캐시 적응을 높이는 동시에 메인 메모리의 쓰기량을 줄이는 방식으로 교체 대

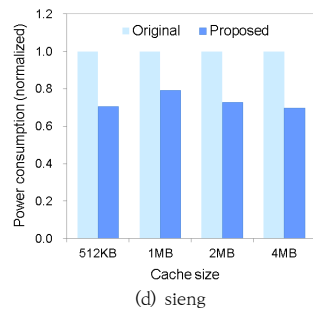
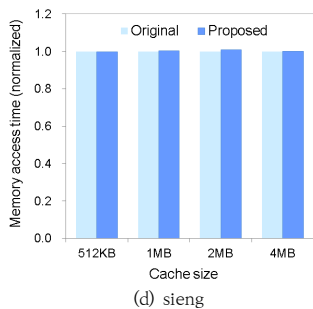
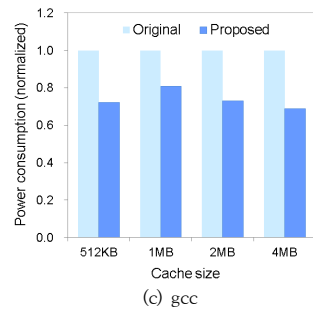
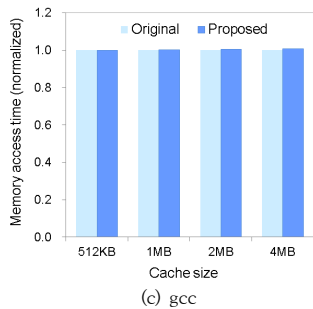
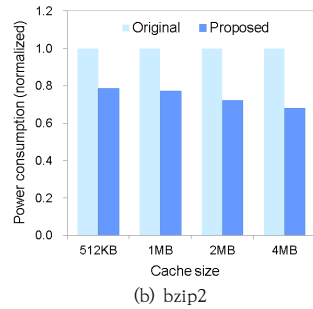
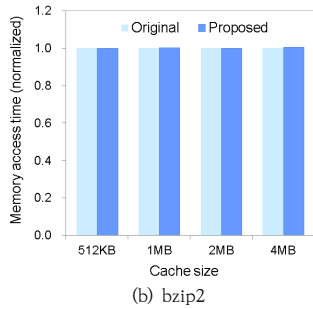
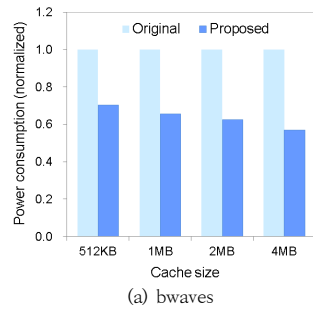
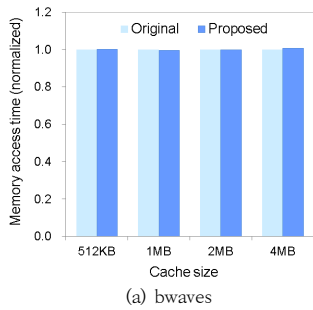


그림 3. 제안 기법에 의한 메모리 접근 시간 효과
Fig. 3. Memory access time by adopting the proposed scheme.

그림 4. 제안 기법에 의한 전력 절감 효과
Fig. 4. Power consumption by adopting the proposed scheme.

상을 선정한다. 중앙처리기에 의해 캐시 블록이 접근될 때 참조 비트가 1로 셋팅되며, 해당 접근이 쓰기 연산에 의한 접근인 경우 쓰기가 발생한 캐시 라인의 수정 비트

를 1로 셋팅한다. 캐시 메모리 내의 모든 캐시 블록들에 대한 참조 비트는 주기적으로 0으로 리셋하여 참조가 최근에 일어났는지 여부를 모니터링할 수 있게 한다. 이는

전통적인 캐시 메모리 관리 기법에서 사용하는 방법과 동일하다. 따라서, 참조 비트가 1인 캐시 블록은 주기적인 리셋 이후 적어도 1번 이상의 참조가 발생했다는 것을 의미하므로 참조 비트는 해당 캐시 블록이 최근에 접근되었는지를 확인하는 식별자로 사용될 수 있다. 즉, 참조 비트를 통해 특정 캐시 블록이 가까운 미래에 다시 참조될 가능성이 높은지를 예측할 수 있다. 캐시 적중을 높이기 위해 본 연구에서는 교체할 캐시 블록을 선정할 때 그 후보군을 참조 비트가 0인 캐시 블록으로 한정한다. 또한 메인 메모리의 쓰기량을 줄이기 위해 이러한 교체 대상 후보군들의 수정 비트를 조사하여 수정된 캐시 라인의 수가 가장 적은 캐시 블록을 최종적인 교체 대상으로 결정한다. 이러한 방식은 최근에 접근된 캐시 블록들은 캐시 메모리에 유지하면서 메인 메모리에 쓰기를 적게 발생시키는 캐시 블록을 교체하는 특성을 가진다.

III. 성능 평가

제안하는 기법의 효과를 검증하기 위해 기존의 캐시 관리 기법과 제안하는 기법을 시뮬레이션을 통해 결과를 비교하였다. 기존 기법은 교체할 캐시 블록 선정시 참조 비트가 0인 캐시 블록을 선택하며, 만약 교체될 캐시 블록의 수정 비트가 세팅된 경우 이를 메인 메모리에 반영한다. 제안하는 기법과의 차이점은 수정 비트를 캐시 라인 단위가 아닌 캐시 블록 단위로 두고 있다는 점과, 교체 대상 선정시 참조 비트가 0인 캐시 블록들은 모두 동일하게 평가하며 수정 비트의 값을 고려하지는 않는다는 점이다. 참고로 이러한 캐시 블록 교체 알고리즘은 현대의 고성능 처리기의 캐시 메모리에서 널리 사용되는 방법 중 하나이다. 본 실험의 기본적인 시스템 구성은 2.4 GHz 듀얼 코어 중앙처리기와 16 방향 연관캐시 방법을 적용하는 최종단 캐시 메모리로 구성되며, 그 크기는 512KB부터 4MB까지 변화할 수 있도록 설정하였다. 최종단 캐시의 캐시 블록은 4KB, 캐시 라인은 256B로 구성된다. 1단 캐시 메모리의 경우 명령 캐시는 16KB, 데이터 캐시는 8KB로 구성되며, 32 바이트 캐시 라인의 직접 사상 방법을 사용하고, 2단 캐시 메모리의 경우 256KB 용량으로 32바이트 캐시 라인의 8 방향 연관 캐시 방법을 사용한다. 메인 메모리로 사용된 상변화 메모리는 읽기/쓰기 지연이 각각 20ns와 100ns로 설정되었으며, 읽기/쓰기 에너지는 각각 0.2nJ/bit와 1.0nJ/bit로 설정되었다. 읽기/쓰기와 무관한 고정된 전력 소모

는 0.1W/GB로 설정되었다. 성능 평가를 위해 실험에 사용한 워크로드는 스펙 벤치마크로 bwaves, bzip2, gcc, sieng의 4가지로 구성된다.

그림 2는 제안한 기법을 사용하기 전후 상변화 메모리에 발생하는 쓰기량을 비교해서 보여주고 있다. 그림에서 보는 것처럼 다양한 캐시 크기와 워크로드 집합에 대해 제안하는 기법은 상변화 메모리의 쓰기량을 크게 줄이는 것을 확인할 수 있으며, 쓰기량 감소폭은 평균 34.6%, 최대 51.6%에 이르는 것을 확인하였다. 그림 3은 제안한 기법의 사용 전후 평균 메모리 접근시간을 보여주고 있다. 그림에서 보는 것처럼 제안하는 기법의 적용으로 인해 상변화 메모리에 발생하는 쓰기량을 크게 줄임에도 불구하고 메모리 성능 저하가 거의 발생하지 않는 것을 확인할 수 있다. 최대 성능 저하가 1% 미만이었으며 심지어 성능이 개선되는 경우도 있었다. 이는 제안하는 기법이 교체 대상인 캐시 블록 선정 시 기존처럼 참조 비트가 0인 동일한 조건을 만족하는 블록으로 제한하기 때문이다. 그림 4는 제안하는 기법 적용 전후의 전력 소모량을 보여주고 있다. 그림에서 보는 것처럼 제안하는 기법의 전력 절감 효과가 상당히 큰 것을 확인할 수 있으며, 그 평균은 28.9%, 최대치는 42.9%에 이르렀다. 이는 상변화 메모리가 읽기 연산에 비해 쓰기 연산의 전력 소모가 상대적으로 크며, 제안하는 기법이 쓰기 연산을 최대한 억제하기 때문에 나타나는 효과로 볼 수 있다.

IV. 결론

본 논문은 차세대 사물인터넷 디바이스를 위한 에너지 효율적인 캐시 및 메모리 관리 기법을 제안하였다. 제안한 기법은 전력 소모가 적은 상변화 메모리를 사물인터넷 디바이스의 메인 메모리로 채택하고 캐시 메모리에서 수정된 블록의 상태를 메인 메모리에 발생시킬 쓰기량에 따라 다단계로 분류한 후 쓰기 발생량이 적은 캐시 블록을 우선적으로 교체하여 쓰기에 취약한 상변화 메모리에 발생하는 쓰기량을 줄인다. 또한, 참조 비트가 1인 캐시 블록은 교체 대상에서 제외함으로써 캐시 메모리의 성능을 훼손하지 않으면서 메모리의 에너지 소모를 줄이는 방식을 채택한다. 스펙 벤치마크를 이용한 시뮬레이션 실험을 통해 제안한 기법이 상변화 메모리에 발생하는 쓰기량을 평균 34.6% 줄이고 전력 소모를 28.9% 줄이면서 메모리의 성능 저하는 발생하지 않음을 보였다.

References

- [1] S. Nam and H. Bahn, "Real-time Task Scheduling Methods to Incorporate Low-power Techniques of Processors and Memory in IoT Environments", The Journal of The Institute of Internet, Broadcasting and Communication, vol. 17, no. 2, pp. 1-5, 2017.
DOI: <https://doi.org/10.7236/JIBC.2017.17.2.1>
- [2] E. Lee, H. Kang, H. Bahn, and K. Shin, "Eliminating Periodic Flush Overhead of File I/O with Non-volatile Buffer Cache," IEEE Trans. Computers, vol. 65, no. 4, pp. 1145-1157, 2016.
DOI: <https://doi.org/10.1109/TC.2014.2349525>
- [3] S. Lee, H. Bahn, S. Noh, "CLOCK-DWF: A Write-History-Aware Page Replacement Algorithm for Hybrid PCM and DRAM Memory Architectures," IEEE Trans. Computers, vol. 63, no. 9, pp. 2187-2200, 2014.
DOI: <https://doi.org/10.1109/TC.2013.98>
- [4] E. Lee, J. Kim, H. Bahn, S. Lee, S. Noh, "Reducing Write Amplification of Flash Storage through Cooperative Data Management with NVM," ACM Trans. Storage, vol. 13, no. 2, 2017.
DOI: <https://doi.org/10.1145/3060146>
- [5] H. Jeon and S. Lee, "Analysis of Remote Update Vulnerabilities of IoT Healthcare Devices," The Journal of KIIT, vol. 19, no. 1, pp.87-97, 2021.
DOI: doi.org/10.14801/jkiit.2021.19.1.87
- [6] J. Park and E. Park, "Performance Evaluation of IoT Cloud Platforms for Smart Buildings," Journal of the Korea Academia-Industrial cooperation Society(JKAIS), vol. 21, no. 5 pp. 664-671, 2020.
DOI: doi.org/10.5762/KAIS.2020.21.5.664
- [7] E. Lee and H. Bahn, "Electricity Usage Scheduling in Smart Building Environments Using Smart Devices," The Scientific World Journal, vol. 2013, article 468097, pp.1-11, 2013.
DOI: <https://doi.org/10.1155/2013/468097>
- [8] D. Kim, E. Lee, S. Ahn, and H. Bahn, "Improving the storage performance of smartphones through journaling in non-volatile memory," IEEE Trans. Consumer Electronics, vol. 59, no. 3, pp. 556-561, 2013.
DOI: doi.org/10.1109/TCE.2013.6626238
- [9] H. Bahn, H. Lee, S. Noh, S. Min, and K. Koh, "Replica-aware Caching for Web Proxies, Computer Communications, vol. 25, no. 3, pp. 183-188, 2002.
DOI: [doi.org/10.1016/S0140-3664\(01\)00365-6](https://doi.org/10.1016/S0140-3664(01)00365-6)
- [10] J. Park, H. Lee, S. Hyun, K. Koh, and H. Bahn, "A

Cost-aware Page Replacement Algorithm for NAND Flash Based Mobile Embedded Systems," In ACM EMSOFT Conf., pp. 315-324, 2009.

DOI: <https://doi.org/10.1145/1629335.1629377>

저 자 소개

반 효 경(정회원)



- 1997년 2월: 서울대학교 계산통계학과 학사
- 1999년 2월: 서울대학교 전산과학과 석사
- 2002년 2월: 서울대학교 컴퓨터공학부 박사.
- 2002년 9월~: 이화여자대학교 컴퓨터공학과 교수.

터공학과 교수.

- 주관심분야: 운영체제, 스토리지시스템, 임베디드시스템

※ This research was funded by the ICT R&D program of MSIP/IITP (2019-0-00074, developing system software technologies for emerging new memory that adaptively learn workload characteristics) and also by the Basic Science Research Program through the NRF grant funded by Korea Government (MSIP) (No. 2019R1A2C1009275).