The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 23, No. 4, pp.53-58, Aug. 31, 2023. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2023.23.4.53
JIIBC 2023-4-9

모바일 기기를 위한 다중 스왑 아키텍처의 설계 및 성능 분석

Design and Performance Analysis of Multi-Swap Architectures for Mobile Devices

반효경*, 김지선**

Hyokyung Bahn*, Jisun Kim**

요 약 스마트폰이 다양한 애플리케이션의 수행을 지원하면서 가상메모리 스와핑 기능의 중요성이 증가하고 있다. 그러나, 전통적인 컴퓨터 시스템과 달리 모바일 플랫폼은 스와핑을 지원하지 않는 것을 기본 옵션으로 채택한다. 그 중요한 이유 중 하나는 스마트폰의 스토리지인 플래시메모리가 스와핑으로 인해 다량의 쓰기 연산 발생시 성능 저하가 일어날 수 있기 때문이다. 본 논문에서는 이러한 문제점을 해결하기 위해 스마트폰을 위한 2가지 다중 스왑 아키텍처인 계층적 스왑과 병렬적 스왑을 제안하고 이들의 성능을 정량적으로 분석한다. 특히, 본 논문에서는 병렬적 스왑 구조에서 단발성 접근 데이터의 특성을 고려한 관리 방식을 제안하여 플래시메모리에 발생하는 스왑량을 줄이고 기존 스왑에 비해 성능 개선 효과가 큼을 입증한다.

Abstract As smartphones increasingly support the execution of various applications, the function of virtual memory swapping is becoming important. However, unlike traditional computer systems, mobile platforms do not basically support swapping. This is because swapping results in frequent writes to flash memory, which may degrade the performance of smartphone's storage significantly. To cope with this situation, this paper suggests two multi-swap architectures, hierarchical swapping and hybrid swapping, and compares their performance quantitatively. Specifically, this paper shows that hybrid swapping with the consideration of single-access data can reduce swapping traffic to flash memory, and improve the performance compared to traditional swapping.

Key Words : Smartphone, swapping, multi-swap, Android, mobile platform, NVM

Ⅰ. 서 론

최근 스마트폰이 제공하는 애플리케이션의 영역이 확대되면서[1, 2, 3] 가상메모리의 스와핑 기능이 점점 중요해지고 있다[4]. 특히, 스마트폰이 금융 거래 등 트랜잭션 단위의 업무를 지원하면서 프로그램 실행의 신뢰성 보장이 필수적인 상황이 되었다.

그러나, 안드로이드 등 현행 스마트폰 플랫폼은 메모리 부족시 스와핑을 지원하는 대신 우선순위가 낮은 프로그램을 강제 종료시키는 방식을 채택하고 있다[5]. 이는

*정회원, 이화여자대학교 컴퓨터공학과
**준회원, 이화여자대학교 컴퓨터공학과
접수일자 2023년 7월 21일, 수정완료 2023년 7월 30일
게재확정일자 2023년 8월 4일

Received: 21 July, 2023 / Revised: 30 July, 2023 /
Accepted: 4 August, 2023
*Corresponding Author: bahn@ewha.ac.kr
Dept. of Computer Engineering, Ewha University, Korea

스마트폰의 스토리지인 플래시메모리가 스와핑 기능을 지원하기에는 대량의 쓰기 연산 발생시 안정적인 성능을 보장하지 못하는 문제에 기인한다. 구체적으로 플래시메모리 스왑을 활성화할 경우 안드로이드 앱의 런치 시간이 2배 이상 느려진다는 보고가 있다[6].

스마트폰의 느린 스와핑 문제를 해결하기 위한 방안으로 소량의 NVM(non-volatile memory)을 함께 사용하는 방안이 고려되고 있다[7, 8]. 특히, 삼성전자의 eMRAM은 모바일 기기에 특화된 NVM으로 제품 개발이 준비중인 상황이다[9]. 초기의 NVM은 고집적성, 저전력성, 바이트단위 접근성 등의 좋은 특징으로 인해 DRAM 메모리의 대체 혹은 보완용으로 연구된 바 있다[10]. 그러나, 일부 NVM 매체들의 경우 메모리로 사용하기에는 느린 접근 시간 및 수명 제한 등으로 스토리지 가속용으로 점차 연구방향이 전환되었다[7, 11]. 이는 메모리가 아닌 스토리지 매체로서의 NVM은 셀의 수명이나 접근 시간 등에서 충분히 경쟁력이 있기 때문이다.

그러나, NVM은 단위 용량당 단가가 높기 때문에 기존 스토리지를 전면 대체하기는 어렵다. 따라서, 스마트폰의 경우 플래시메모리로 구성된 스토리지의 성능 가속용으로 소량의 NVM을 추가 구성하는 방안이 대안이 될 수 있다. 본 논문에서는 스마트폰 시스템의 스와핑 지원을 위한 다중 스왑 장치 아키텍처를 제안하고 그 성능을 비교 분석한다.

구체적으로는 NVM과 플래시메모리로 구성된 다중 스왑 아키텍처 2종을 제안하고 이들의 효율적인 관리 방안을 모색한다. 첫 번째 아키텍처는 그림 1(a)에 보는 것처럼 NVM과 플래시메모리를 계층적으로 배치하는 방안으로 이러한 구조에서는 페이지 폴트 발생 시 NVM이 먼저 탐색되고 해당 데이터가 NVM에 없을 때 플래시메모리가 접근된다. 두 번째 구조는 그림 1(b)에서 보는 것처럼 NVM과 플래시메모리를 동일 수준의 스왑 장치로 구성하고 병렬적으로 관리하는 방안이다.

본 논문은 병렬적 스왑 구조를 위한 효율적인 관리 기법을 제안한다. 제안하는 기법은 스마트폰의 스왑 데이터 참조 특성을 고려해서 비인기 데이터가 NVM에 저장되는 것을 방지하여 전체적인 플래시메모리 접근 횟수를 줄인다. 특히, 안드로이드의 스왑 데이터 참조 특성 분석을 통해 많은 양의 단발성 스왑 참조가 발생하는 것을 확인하고 이를 NVM에 진입하지 못하도록 설계하여 소량의 NVM 도입에 의한 효과를 극대화한다. 본 논문에서는 병렬적 스왑 구조와 계층적 스왑 구조의 성능을 비교 분석하고 병렬적 스왑 구조에 제안한 관리 기법을 적용할 경우 모바일 기기의 스와핑을 충분히 좋은 성능으로 지원할 수 있음을 보인다.

본 논문의 이후 구성은 다음과 같다. II장에서는 스마트폰을 위한 다중 스왑 아키텍처를 설명하고 본 논문이 제안하는 관리 기법을 설명한다. III장에서는 시뮬레이션 및 실측 실험을 통해 다양한 스왑 아키텍처의 성능을 비교 분석한다. 끝으로 IV장에서는 본 논문의 결론을 제시한다.

II. 스마트폰을 위한 다중 스왑 구조

본 장에서는 스마트폰의 스와핑 지원을 위한 NVM과 플래시메모리의 다중 스왑 구조를 설명하고 효율적인 관리 기법을 제안한다. 그림 1(a)에 표시된 계층적 스왑 구조는 통상적인 메모리/스토리지 구조에서 매체의 용량과 접근 속도 차이가 발생할 때 가장 널리 사용되는 구조이다. 이 방식의 장점은 NVM이 일종의 캐쉬 역할을 하므로 자주 요청되는 스왑 데이터의 경우 NVM을 통해 처리되고 플래시메모리까지 요청이 전달되지 않는다는 점이다. NVM의 공간 제약으로 데이터 삭제가 필요한 경우 최근에 사용되지 않은 데이터를 쫓아내는 LRU(least recently used) 또는 CLOCK 알고리즘을 사용하는 것이 일반적이다. 한편, 이 방식은 NVM의 용량이 작고 NVM에 진입 후 재사용되지 않는 데이터가 많은 경우 비효율성을 초래할 수 있다. 이러한 유형의 데이터도 시간이 흐르면 NVM에서 쫓겨나지만 소량의 NVM 공간을 오염시켜 빈번히 사용되는 데이터가 쫓겨난 후 재진입하는 등 플래시메모리의 접근을 늘어나게 하는 경우가 발생할 수 있다.

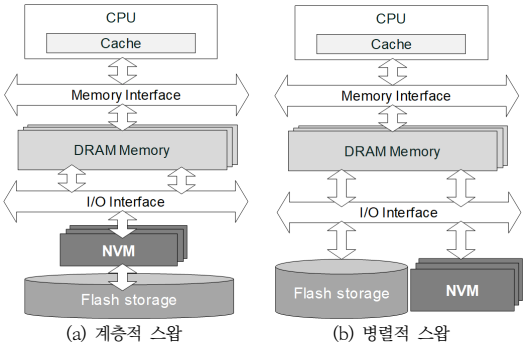

그림 1. 다중 스왑 아키텍처의 비교
Fig. 1. Comparison of multi-swap architectures.

그림 1(b)에 표시된 병렬적 스왑 구조는 인기 데이터와 비인기 데이터를 각각 NVM과 플래시메모리에 저장하여 효율성을 높일 수 있지만, 데이터 자체를 효율적으로 배치하기 위한 관리 방법이 필요하다. 한편, 본 논문에서는 스와핑 지원 안드로이드 기기에서 발생하는 스토리지 접근을 단발성 접근과 다중 접근으로 구분해서 분석한 결과 단발성 접근이 평균 50%를 상회하는 것을 확인하였다. 이러한 대량의 단발성 접근 데이터의 경우 고속 스왑 장치에 배치하더라도 성능 개선 효과가 없기 때문에 이들을 식별하여 NVM에 배치하지 않는 것이 중요하다. 이는 고속 스왑장치인 NVM에 배치될 데이터를 결정하기 위해 해당 데이터가 단발성 접근 데이터인지를 확인 또는 예측하는 절차가 필요함을 뜻한다.

본 논문에서는 메모리 페이지마다 존재하는 접근 비트의 기능을 스왑 장치 관리로 확장하여 스와핑 대상 데이터가 최근에 접근되었는지를 확인하는 용도로 사용하고 페이지 폴트 발생시 그 데이터의 위치를 2종의 스왑장치에서 탐색한다. 해당 데이터가 플래시메모리에 존재하는 경우 접근 비트가 이미 1이었으면 이를 NVM으로 승격시킨다. 이때, NVM에 빈 공간이 없는 경우 NVM에서 최근에 접근되지 않은 데이터를 탐색하고 이를 플래시메모리로 강등시킨다. 최근에 접근되지 않은 데이터를 선택하기 위해서는 CLOCK 알고리즘을 사용하여 NVM 내의 데이터를 순환적으로 탐색하며 접근 비트가 1인 데이터는 0으로 바꾸고 접근 비트가 0인 데이터를 선택한다. 한편, 메모리에서 처음 스왑 아웃되는 데이터는 플래시메모리에 위치시킨다. 이러한 방식을 통해 NVM은 빈번히 요청되는 인기 데이터를 보관하는 용도로 사용되며, 그렇지 않은 데이터는 접근 비트를 통해 진입을 차단하게 된다.

III. 성능 평가

본 장에서는 다중 스왑 구조 및 관리 방식에 따른 성능 평가를 수행한다. 이를 위해 먼저 병렬적 스왑 구조와 계층적 스왑 구조를 비교하는 실험을 수행한다. 그런 다음 병렬적 스왑 구조 상에서 제안한 알고리즘과 기존의 CLOCK 알고리즘으로 메모리 관리를 수행한 경우 성능을 비교 분석한다. 끝으로, 플래시메모리만을 스왑 장치로 사용하는 구조, 스와핑 자체를 사용하지 않는 시스템과도 다중 스왑 구조의 성능을 비교한다.

본 논문의 실험을 위해 안드로이드의 스왑 입출력 트레이스를 추출하고 이를 재현하는 시뮬레이션을 수행하였다. 실험에서는 20종의 앱으로 다양한 시나리오를 구성한 후 각각 앱들의 실행 시퀀스를 반복하여 스와핑을 발생시키고 그에 따른 입출력이 발생하도록 하였다. 이때 스왑 파일은 /mnt/sdcard/ 파티션에 생성되도록 하였고 트레이스 수집을 위해서는 ftrace를 사용하였다.

그림 2는 병렬적 스왑 구조와 계층적 스왑 구조를 사용했을 때 스토리지 입출력 시간을 비교해서 보여주고 있다. 두 방식 모두 NVM 공간 부족으로 인한 데이터 교체 상황 발생시 CLOCK 알고리즘을 사용했으며, 병렬적 스왑에서는 데이터의 스왑 위치를 NVM과 플래시메모리 중 결정하기 위해 제안하는 알고리즘을 사용했다. 그림

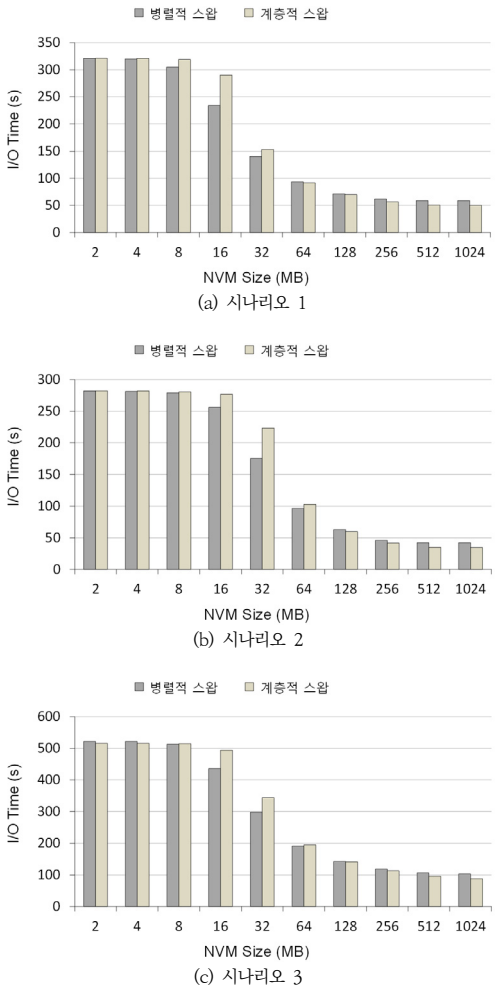

그림 2. 다중 스왑 아키텍처에 따른 성능 비교
Fig. 2. Performance comparison as the multi-swap architecture is varied.

에서 보는 것처럼 병렬적 스왑이 계층적 스왑에 비해 입출력 시간이 일정구간에서 개선된 것을 확인할 수 있다. 구체적인 성능 개선 폭은 평균 5% 최대 22%에 이르렀다. 이는 병렬적 스왑이 계층적 스왑보다 NVM 크기가 소규모인 구간에서 더 우수한 방식임을 뜻한다. 병렬적 스왑은 데이터가 최초로 스왑 장치에 저장될 때 NVM이 아닌 플래시메모리에 저장되도록 하여 비인기 데이터의 NVM 진입을 차단한다. 이에 비해 계층적 스왑은 모든 데이터를 NVM에 삽입하므로 인기 데이터가 비인기 데이터에 의해 밀려나는 상황이 발생한다. 이는 병렬적 스왑이 모바일 플랫폼의 스왑 참조 성향을 계층적 스왑보다 더 잘 반영했기 때문으로 볼 수 있다. 그러나, NVM의 용량이 충분히 큰 경우에는 계층적 스왑이 더 우수한 성능을 나타내었다. 이는 병렬적 스왑이 NVM 공간에 여유가 있는 상황에서도 최초의 스왑 진입을 플래시메모리로 설정하여 인기 데이터의 경우 플래시메모리 접근이 1회 추가되기 때문이다.

그림 3은 병렬적 스왑 구조에서의 CLOCK 알고리즘과 본 논문에서 제안한 알고리즘의 플래시메모리 접근 횟수를 비교해서 보여주고 있다. 그림에서 보는 것처럼 제안한 알고리즘은 CLOCK과 비교해서 플래시메모리 접근 횟수가 일부 구간에서 줄어든 것을 확인할 수 있다. 이는 제안한 알고리즘과 달리 CLOCK이 단발성 접근에 대한 NVM의 진입 통제를 하지 않아 적은 크기의 NVM이 단발성 접근 데이터에 의해 오염되고 다중 접근 데이터를 몰아내는 역할을 하기 때문이다. 제안한 알고리즘의 CLOCK 대비 플래시메모리 접근 횟수는 평균 6% 최대 24%까지 줄어든 것을 확인할 수 있었다. 그러나, NVM의 크기가 충분히 큰 경우에는 두 알고리즘이 유사한 결과를 나타냈으며, 일부 경우에는 CLOCK이 근소하게 우위를 나타내는 경우도 있었다. 이는 CLOCK이 단발성 접근과 다중 접근 데이터를 구분없이 NVM에 곧바로 진입시키는 반면 제안한 알고리즘은 NVM의 여유 공간이 있는 경우에도 단발성 접근 데이터의 진입을 막기 때문이다.

그림 4는 두 알고리즘의 NVM 쓰기 횟수를 비교해서 보여주고 있다. 그림에서 보는 것처럼 제안한 알고리즘이 CLOCK 대비 NVM 쓰기 횟수를 크게 줄였음을 확인할 수 있다. 이는 제안 알고리즘이 다중 접근 데이터를 NVM에 진입시킨 후 계속 유지하고 있는 반면 CLOCK

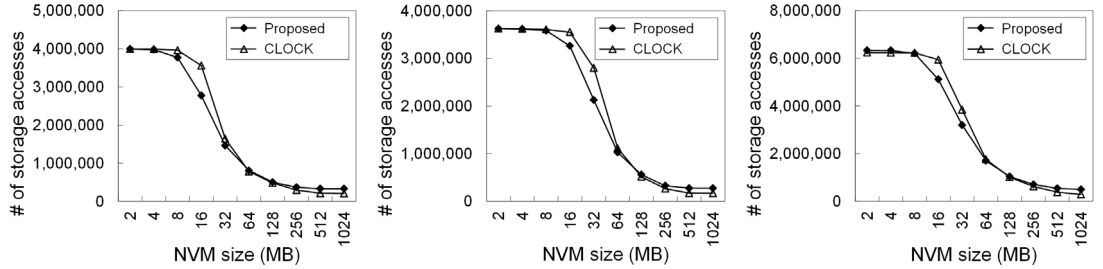

그림 3. CLOCK 알고리즘과 제안하는 알고리즘의 스토리지 접근 횟수 비교
Fig. 3. Comparison of storage access counts when CLOCK and the proposed algorithm is used

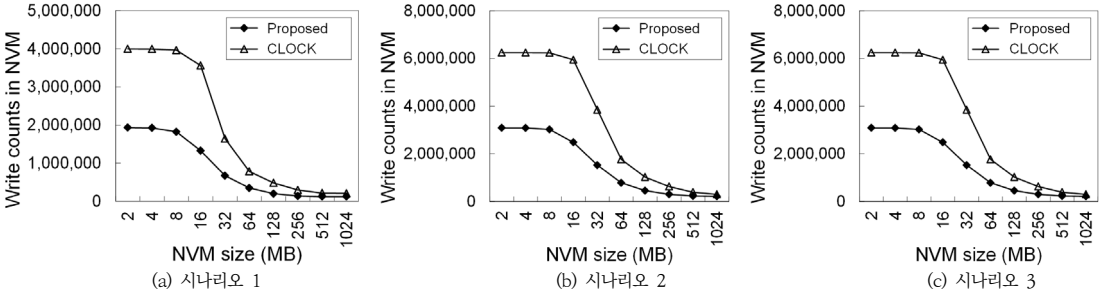

그림 4. CLOCK 알고리즘과 제안하는 알고리즘의 NVM 쓰기 횟수 비교
Fig. 4. Comparison of NVM write counts when CLOCK and the proposed algorithm is used

은 이미 저장된 다중 접근 데이터가 단발성 접근 데이터의 진입에 의한 공간 부족으로 쫓겨나고 재진입하는 과정에서 많은 양의 NVM 쓰기를 발생시키기 때문이다. 제안한 알고리즘은 평균 55%, 최대 64%의 NVM 쓰기 감소 효과가 있었으며, 이는 쓰기에 취약한 NVM 매체 특성상 유의미한 성과로 볼 수 있다.

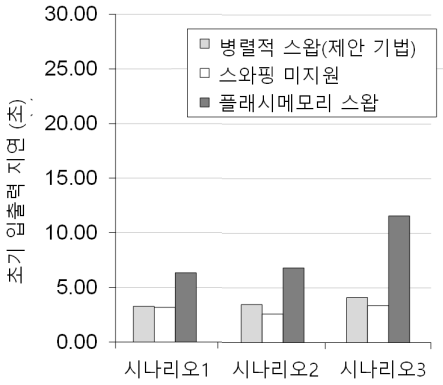

그림 5. 초기 입출력 지연 시간 비교
Fig. 5. Comparison of launch time

그림 5는 본 논문의 병렬적 스왑과 NVM을 사용하지 않는 플래시메모리 스왑, 그리고 스와핑 자체를 사용하지 않는 시스템의 앱별 평균 런치 시간을 비교해서 보여주고 있다. 그림에서 보는 것처럼 병렬적 스왑에 제안한 알고리즘을 적용한 경우가 플래시메모리 스왑 대비 평균 53%의 성능 개선이 나타난 반면 스와핑 자체를 사용하지 않는 시스템과 비교해서는 그다지 성능 저하의 폭이 크지 않음을 확인할 수 있었다.

IV. 결 론

본 논문에서는 플래시메모리와 NVM으로 구성되는 모바일 기기용 다중 스왑 시스템의 아키텍처인 병렬적 스왑과 계층적 스왑의 성능을 비교하고 병렬적 스왑을 위한 효율적인 관리 방안을 제안하였다. 특히 본 논문은 단발성 접근이 잦은 스왑 데이터 참조의 특성을 반영하여 인기 데이터의 스왑 참조를 NVM이 최대한 흡수하고 비인기 데이터의 NVM 진입을 막는 정책을 제안하여 기존의 CLOCK 알고리즘 대비 평균 6% 최대 24%의 성능 개선 효과를 얻었으며, 계층적 스왑과 비교해서는 평균 5% 최대 22%의 성능 개선을 얻었다.

References

[1] I. Nayeem and R. Want, "Smartphones: past, present, and future," IEEE Pervasive Computing, vol. 13, no. 4, pp. 89-92, 2014
DOI: https://doi.org/10.1109/MPRV.2014.74.

[2] B. Lee and C. Son, "Improving evaluation metric of mobile application service with user review data," Journal of the Korea Academia-Industrial cooperation Society, vol. 21, no. 1, pp. 380-386, 2020.
DOI: https://doi.org/10.5762/KAIS.2020.21.1.3

[3] B. Choi, S. Eom, C. Kim, and H. Lee, "Counterfeit bill identification based on deep learning using smartphone camera shooting images," Journal of KIIT, vol. 19, no. 3, pp. 1-8, 2021.
DOI: https://doi.org/10.14801/jkiit.2021.19.3.1

[4] S. Yoon, H. Park, K. Cho, and H. Bahn, "Supporting swap in real-time task scheduling for unified power-saving in CPU and memory," IEEE Access, vol. 10, pp. 3559-3570, 2022.
DOI: https://doi.org/10.1109/ACCESS.2021.3140166

[5] S. Kim, J. Jeong, J. Kim, and S. Maeng, "SmartLMK: a memory reclamation scheme for improving user-perceived app launch time," ACM Trans. Embed. Comput. Syst., vol. 15, no. 47, pp. 1-25, 2016.
DOI: https://doi.org/10.1145/2894755

[6] J. Kim and H. Bahn, "Analysis of smartphone I/O characteristics — toward efficient swap in a smartphone," IEEE Access, vol. 7, pp. 129930-129941, 2019.
DOI: https://doi.org/10.1109/ACCESS.2019.2937852.

[7] H. Bahn and K. Cho, "Implications of NVM based storage on memory subsystem management," Applied Sciences, vol. 10, no. 3, 2020.
DOI: https://doi.org/10.3390/app10030999

[8] Y. Park and H. Bahn, "Modeling and analysis of the page sizing problem for NVM storage in virtualized systems," IEEE Access, vol. 9, pp. 52839-52850, 2021.
DOI: https://doi.org/10.1109/ACCESS.2021.3069966

[9] K. Suh, J. Lee, H. Shin, J. Lee et al., "12.5 Mb/mm2 Embedded MRAM for High Density Non-volatile RAM Applications," IEEE Symp. VLSI Technology, 2021.

[10] S. Lee, H. Bahn, and S. H. 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

[11] H. Bahn and Y. Park, "Workload-aware page size modeling for fast storage in virtualized environments," The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), vol. 22, no. 3, pp. 93-98, 2022.
DOI: https://doi.org/10.7236/JIIBC.2022.22.3.93

저 자 소 개

반 효 경(정회원)

- 1997년 : 서울대학교 계산통계학과 학사
- 1999년 : 서울대학교 전산과학과 석사
- 2002년 : 서울대학교 컴퓨터공학부 박사.
- 2002년 ~ : 이화여자대학교 컴퓨터공학과 교수.
- 주관심분야 : 운영체제, 스토리지시스템, 임베디드시스템

김 지 선(준회원)

- 2011년 2월 : 한신대학교 컴퓨터공학과 학사
- 2014년 3월 ~ : 이화여자대학교 컴퓨터공학과 대학원생
- 주관심분야 : 운영체제, 스토리지 시스템

※ This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub) and (No.RS-2022-00155966, Artificial Intelligence Convergence Innovation Human Resources Development (Ewha University)).