The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 22, No. 6, pp.57-62, Dec. 31, 2022. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2022.22.6.57
JIIBC 2022-6-9

스토리지 쓰기량과 페이지 폴트를 줄이는 메모리 부하 적응형 페이지 교체 정책

Page Replacement Policy for Memory Load Adaption to Reduce Storage Writes and Page Faults

반효경*, 박윤주**

Hyokyung Bahn*, Yunjoo Park**

요 약 최근 상변화메모리와 같은 고속 스토리지 매체의 출현으로 느린 디스크 스토리지에 적합하게 설계된 메모리 관리 기법에 대한 재고가 필요한 시점에 이르렀다. 본 논문에서는 상변화메모리를 가상메모리의 스왑장치로 이용하는 시스템을 위한 새로운 페이지 교체 정책을 제안한다. 제안하는 방식은 페이지 교체 정책이 전통적으로 추구하던 페이지 폴트 횟수 절감뿐 아니라 스왑 장치에 발생하는 쓰기량 절감을 동시에 추구한다. 이는 상변화메모리의 쓰기 연산이 느리고 쓰기 횟수에 제한이 있다는 점에 착안한 것이다. 구체적으로 살펴보면 메모리 부하가 높은 경우 페이지 폴트를 줄이는 데에 초점을 맞추고 메모리 공간에 여유가 있을 경우 스토리지 쓰기량을 줄이는 적응적인 방식을 채택한다. 이를 통해 제안하는 정책이 메모리 시스템의 성능을 저하시키지 않으면서 스토리지 쓰기량을 크게 절감함을 다양한 워크로드의 메모리 참조 트레이스를 재현하는 시뮬레이션 실험을 통해 보인다.

Abstract Recently, fast storage media such as phage-change memory (PCM) emerge, and memory management policies for slow disk storage need to be revisited. In this paper, we propose a new page replacement policy that makes use of PCM as a swap device of virtual memory systems. The proposed policy aims at reducing write traffic to the swap device as well as reducing the number of page faults pursued by traditional page replacement policies. This is because a write operation in PCM is slow and PCM has limited write endurances. Specifically, the proposed policy focuses on the reduction of page faults when the memory load of the system is high, but it aims at reducing write traffic to storage when free memory space is sufficient. Simulation experiments with various memory reference traces show that the proposed policy reduces write traffic to PCM without performance degradations.

Key Words : Page fault, phase-change memory, replacement policy, virtual memory, CLOCK

*정회원, 이화여자대학교 컴퓨터공학과
**정회원, 이화여자대학교 컴퓨터공학과
접수일자 2022년 11월 6일, 수정완료 2022년 11월 30일
게재확정일자 2022년 12월 9일

Received: 6 November, 2022 / Revised: 30 November, 2022 / Accepted: 9 December, 2022
*Corresponding Author: bahn@ewha.ac.kr
Dept. of Computer Engineering, Ewha University, Korea

Ⅰ. 서 론

전통적으로 메모리 관리 시스템 설계 시 가장 중요하게 생각해 온 요소는 DRAM과 하드 디스크 사이의 큰 속도 차이이다[1, 2]. 느린 스토리지 접근 오버헤드를 줄이기 위해 그간 캐싱(caching) 및 페이지 폴트(page fault) 개선을 위한 많은 연구가 이루어져 왔다[3, 4]. 한편, 최근에는 플래시메모리, 상변화메모리 등 고속 스토리지 기술이 발전하면서 두 계층 간의 속도 차이가 크게 줄어들고 있다[5, 6]. 플래시메모리는 DRAM 메모리와 스토리지 간의 접근 속도 차이를 수천배 이내로 줄였으며, 상변화메모리 등 차세대 고속 스토리지의 출현으로 이러한 접근 속도 차이는 수배에서 수십배 정도까지 줄어들게 되었다[6, 7].

초창기에는 상변화메모리가 DRAM 대비 낮은 전력소모와 고집적도, 바이트 단위 접근성 등의 장점으로 메인 메모리 매체로 활용될 가능성이 대두되었다[8]. 그러나, DRAM과의 성능 격차가 좁혀지지 않고, 특히 느린 쓰기 연산과 쓰기 횟수 제한 등으로 이제는 배터리 제약이 있는 임베디드 시스템용 저전력 메모리로 사용되는 경우[9, 10] 외에는 주로 고속 스토리지로서의 활용 가능성이 더 주목받고 있는 상황이다[6, 11].

본 논문에서는 상변화메모리를 고속 스토리지로 탑재하고 이를 가상메모리의 스왑장치로 사용하는 시스템을 위한 새로운 메모리 페이지 교체 정책을 제안한다. 그림 1은 이러한 고속 스토리지를 탑재한 시스템의 전체적인 시스템 구조를 보여주고 있다.

비록 상변화메모리가 스토리지로서의 고속성과 바이트 단위 접근성을 제공하지만, 쓰기 연산에 취약성을 가지기 때문에 이러한 아키텍쳐 상에서는 가급적 스토리지의 쓰기 연산을 줄이는 메모리 관리 정책이 필요하다[12]. 이를 위해 본 논문에서는 각 메모리 페이지의 수정 여부를 서브페이지 수준에서 관리하면서 스토리지에 많은 쓰기량을 유발하는 페이지의 방출을 최대한 유예하는 방식을 제안한다. 즉, 방출 대상 페이지를 선정할 때 비록 최근에 사용되지 않은 페이지라 하더라도 수정 정도가 높은 페이지의 경우 그 수정 정도에 비례해 메모리로부터의 방출을 유예한다.

한편, 최근에 사용되지 않은 페이지를 지나치게 오래 메모리에 머무르게 할 경우 메모리 공간이 부족한 상황에서는 페이지 폴트를 증가시키는 부작용을 초래할 위험성이 있다. 따라서, 제안하는 정책에서는 메모리 부하를 모니터링하여 수정 페이지의 방출 유예 여부를 적응적으로 조절한다. 다양한 가상메모리 참조 트레이스를 재현하는 시뮬레이션 실험을 통해 제안한 정책이 페이지 폴트의 증가 없이 상변화메모리의 쓰기량을 크게 감소시킬 수 있음을 보인다.

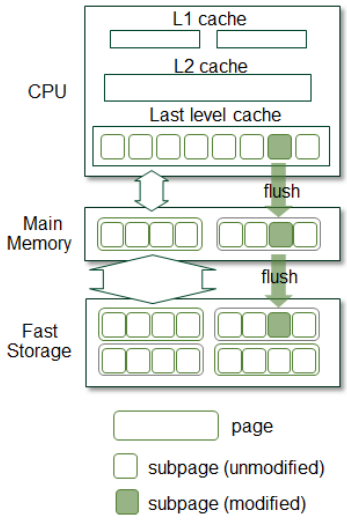


그림 1. 고속 스토리지를 탑재한 가상메모리 구조
Fig. 1. Virtual memory architecture with fast storage.

본 논문의 이후 구성은 다음과 같다. II장에서는 상변화메모리를 스왑장치로 사용하는 시스템을 위해 제안하는 페이지 교체 정책에 대해 상세히 기술한다. III장에서는 시뮬레이션 실험을 통해 제안한 정책에 대한 유효성을 검증한다. 끝으로 IV장에서는 본 논문의 결론을 제시한다.

Ⅱ. 제안하는 페이지 교체 정책

그림 1에서 보는 것처럼 제안하는 시스템 구조에서 메인메모리는 DRAM으로 구성되며, 가상메모리용 스왑 장치로는 고속 스토리지인 상변화메모리가 탑재된다. 메인메모리와 상단의 캐시메모리 간에는 서브페이지 단위로 데이터가 전송되며, 메인메모리와 하단의 스토리지 간에는 페이지 단위로 데이터가 전송된다.

각 페이지에는 참조 비트를 두어 해당 페이지가 최근에 접근되었는지를 표시하며, 또한 각 페이지에는 수정 비트를 두어 해당 페이지가 메인메모리에 올라온 후 수정된 부분이 있는지를 표시한다. 참조 비트는 운영체제가 메모리로부터 방출시킬 페이지를 선정할 때 참고하

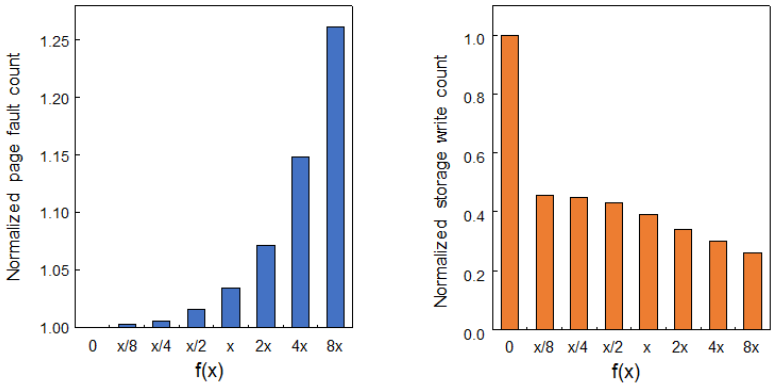
며, 수정 비트는 방출 대상으로 선정된 페이지를 메모리에서 쫓아낼 때 해당 페이지를 스왑 장치에 반영할 필요가 있는지를 결정할 때 활용한다[8, 13]. 이때, 기존 시스템이 페이지별로 수정 비트를 두는 것과 달리 제안하는 알고리즘은 서브페이지 단위로 수정 비트를 두어 해당 페이지가 방출될 때 발생할 스토리지 쓰기량을 예측한다. 서브페이지에 대한 수정 비트는 캐시메모리로부터 메인메모리로 쓰기 연산이 수행될 때 세팅된다. 수정 비트가 세팅된 서브페이지는 이를 포함한 페이지가 메인메모리에서 방출될 때 스왑 장치에 반영된다.

제안하는 알고리즘은 메모리에서 방출시킬 페이지를 결정할 때 각 페이지의 참조 비트와 수정 비트들을 참고하여 페이지 폴트와 스토리지 쓰기량을 동시에 줄인다. 전통적인 클럭 알고리즘과 유사하게 메모리에 여유공간이 필요할 경우 메모리 내의 페이지들로 구성된 환형 리스트에서 클럭 핸드를 시계 방향으로 이동시키며 방출 대상페이지를 선정한다. 각 페이지의 참조 비트는 해당 페이지가 읽기 또는 쓰기 참조 발생시 하드웨어적으로 세팅되며, 각 서브페이지의 수정 비트는 해당 서브페이지에 쓰기 참조 발생시 하드웨어적으로 세팅된다.

따라서, 운영체제는 메모리 공간 확보가 필요한 경우 이러한 비트들을 조사하여 방출할 페이지를 결정한다. 먼저 클럭 핸드가 가리키는 페이지의 참조 비트가 1로 세팅된 경우 이를 0으로 변경하고, 클럭 핸드를 다음 페이지로 전진시킨다. 만약 참조 비트가 0인 경우 해당 페이지에 속한 서브페이지들의 수정 비트들을 확인한다. 제안하는 알고리즘은 비록 참조 비트가 0인 페이지라도 수정 비트가 세팅된 페이지라면 메모리로부터 곧바로 방출하지 않고 일정 기간 유예한다. 이를 위해 각 페이지에 대해 유예 횟수를 유지하며, 이 값이 유예 임계치를 넘지 않을 경우 해당 페이지를 방출하지 않고 클럭 핸드를 다음 페이지로 이동시킨다. 이러한 작업은 방출 대상 페이지가 발견될 때까지 지속된다.

한편, 제안하는 기법은 페이지 내에 세팅된 수정 비트의 수에 따라 유예 임계치를 다르게 설정한다. 즉, 수정 비트가 상대적으로 더 많이 세팅된 페이지의 경우 유예 임계치를 더 높게 책정한다. 이를 위해 함수 f를 두어 해당 페이지 내에서 수정된 서브페이지의 수를 x라 할 때 유예 임계치를 $f(x)$로 표시한다. 스토리지에 발생하는 쓰기량을 줄이기 위해 f는 단조증가함수로 정의하여 쓰기량을 많이 발생시킬 페이지를 메모리에 더 오래 유지한다.

그러나, 메모리 용량이 워크로드 크기에 비해 부족할 경우 이러한 방식은 수정 페이지를 메모리에 지나치게 오래 유지하여 페이지 폴트를 지나치게 증가시킬 우려가 있다. 메모리 상태에 따른 적절한 유예 횟수를 결정하기 위해 본 논문에서는 메모리 참조 트레이스를 재현하는 선행 실험을 통해 그림 2와 같은 결과를 얻었다. 그림 2(a)와 2(b)는 함수 f의 정의가 변함에 따른 페이지 폴트 수와 스토리지 쓰기횟수를 보여주고 있다. 그림에서 $f(x)$=0인 경우는 기존의 클럭 알고리즘과 동일하게 동작하여 수정된 페이지라도 참조 비트가 0인 경우 방출을 유예하지 않으며, $f(x)$가 증가함에 따라 유예 임계치를 높게 설정하여 수정 페이지의 방출을 여러 차례 유예함을 뜻한다. 그림에서 보는 것처럼 $f(x)$가 증가할수록 페이지 폴트는 증가하는 반면 스토리지에 발생하는 쓰기 횟수는 감소하는 것을 확인할 수 있다.



(a) 페이지 폴트 수 (b) 스토리지 쓰기횟수

그림 2. 유예 임계치 함수의 정의에 따른 페이지 폴트 수와 스토리지 쓰기횟수
Fig. 2. Page faults and storage writes as the deferring threshold function is defined differently.

제안하는 기법에서는 메모리 성능을 저하시키지 않는 선에서 스토리지 쓰기량을 줄이기 위해 워크로드에 의한 메모리 부하상태를 모니터링하고 이에 근거해 적절한 유예 임계치를 결정한다. 예를 들어 현재 메모리 부하가 높은 경우 유예 임계치를 낮게 책정하여 페이지 폴트를 감소시키는 데에 초점을 맞추며, 메모리 여유 공간이 충분한 상태에서는 유예 임계치를 높여 스토리지에 발생하는 쓰기량 감소에 초점을 맞춘다.

이러한 시스템 상태 예측을 위해 현재 메모리 크기와 추가적인 메모리가 할당된 가상의 상태에서의 페이지 부재율을 비교해서 현재 메모리가 부족한 상태인지 충분한 상태인지를 파악한다. 이를 위해 본 논문은 메모리 크기가 주어졌을 때 적중률(hit ratio) 예측에 널리 사용되는 Belady의 생애함수(lifetime function)를 사용하였다[12]. 본 논문에서는 적중률 대신 페이지 폴트율을 예측할 수 있도록 주어진 페이지 개수 x에 대한 페이지 폴트율 함수 $A(x)$를

$$A(x) = c * x^{-k} \quad (1)$$

로 정의하였다. 이때, c와 k는 워크로드의 지역성에 따라 결정되는 제어 파라미터로 c의 값이 작아지거나 k의 값이 커질 경우 워크로드의 지역성이 높다는 것을 의미한다.

Ⅲ. 성능 평가

본 논문에서 제안한 페이지 교체 정책의 성능평가를 위해 메모리 참조 트레이스를 재현하는 시뮬레이션 실험을 수행하였다. 트레이스는 Valgrind 툴셋의 Cachegrind를 통해 추출하였으며[14], 리눅스 상의 5종 워크로드인 game, editor, photo, PDF, media와 이들의 조합인 mix1, mix2, mix3로 구성된다. 실험에서 페이지의 크기는 4KB로 설정하였으며, 서브페이지는 512 바이트로 하여 페이지 하나가 총 8개의 서브페이지로 구성되도록 하였다.

그림 3은 제안하는 기법에서 시간이 흐름에 따라 시스템 내의 메모리 상태가 변할 경우 함수 f를 어떻게 조절하고 있는지를 보여주고 있다. 이 실험에서는 시간이 흐름에 따라 시스템 내에서 수행 중인 워크로드의 수를 1개부터 4개까지 변화시켰으며, 이에 따라 함수 $f(x)$가 메모리 여유 공간이 어느 정도인지를 모니터링하고 이에 근거해 유예 임계치를 조절하는 것을 확인할 수 있었다. 구체적으로 살펴보면, 메모리 부하가 높은 논리시간 90

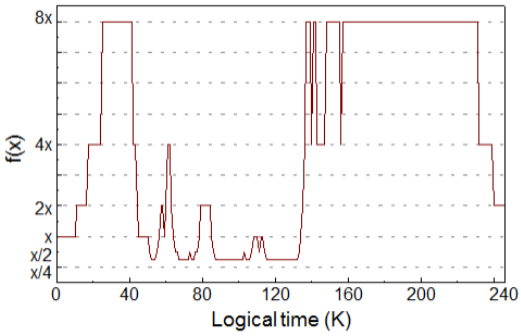
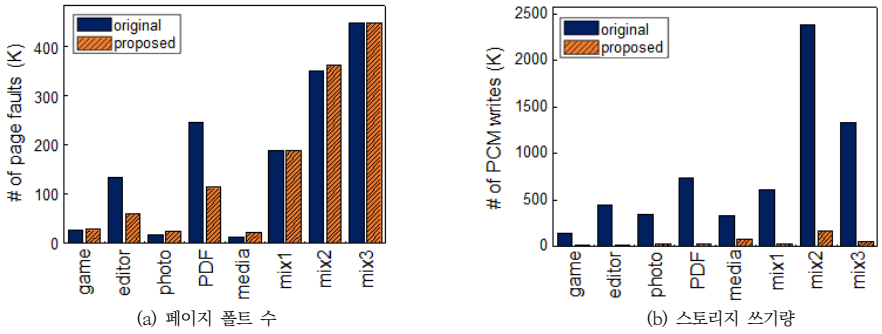


그림 3. 워크로드 변동에 따른 유예 임계치 함수의 적응 과정
Fig. 3. Adaptation of deferring threshold function as the workload is varied.

~130 구간의 경우 $f(x)$를 낮게 설정해서 유예 임계치를 낮추는 것을 확인할 수 있었으며, 메모리 부하가 낮은 논리시간 20~40 및 160~220의 경우 $f(x)$를 높게 설정해서 유예 임계치를 높이는 것을 확인할 수 있다. 이때, 메모리 부하의 높고 낮음에 대한 모니터링은 II장의 페이지 폴트율 함수 $A(x)$를 통해 하게 되며, 페이지 개수 변화에 페이지 폴트율의 변동성이 큰 경우 메모리 부하가 높은 것으로 판별하고 그렇지 않은 경우 메모리 부하가 낮은 것으로 판별한다.

그림 4는 8종의 워크로드에 대해 $f(x)$를 0으로 고정시킨 기존 클럭 알고리즘(original)과 제안하는 정책의 페이지 폴트수와 스토리지 쓰기량을 비교해서 보여주고 있다. 그림에서 보는 것처럼 제안하는 정책이 기존 클럭 알고리즘에 비해 스토리지의 쓰기량을 크게 줄이는 것을 확인할 수 있다. 이는 수정 정도가 높은 페이지의 메모리 방출을 최대한 유예함으로 얻게 된 결과로 분석할 수 있다. 그럼에도 불구하고 제안하는 정책은 기존 알고리즘 대비 페이지 폴트 수를 증가시키지 않는 것을 확인할 수 있다. 오히려 일부 워크로드에서는 페이지 폴트 수를 더 줄이는 경우를 확인할 수 있었는데, 이는 시스템의 메모리 상황에 적합한 관리 정책을 설정함으로 인해 얻게 된 효과로 볼 수 있다.

요약하자면 제안하는 정책은 메모리 용량이 부족할 경우 메모리 성능 개선에 초점을 맞춘 운영을 하며, 메모리 공간에 여유가 있을 때에는 수정된 페이지의 스토리지 방출을 최대한 유예하여 스토리지에 발생하는 쓰기량을 줄이는 성과를 확인할 수 있었다.



(a) 페이지 폴트 수 (b) 스토리지 쓰기량

그림 4. 워크로드에 따른 페이지 폴트 수와 스토리지 쓰기량 비교
Fig. 4. Comparison of page faults and storage writes as the workload is varied.

Ⅳ. 결 론

본 논문에서는 고속 스토리지인 상변화메모리가 스왑 장치로 탑재된 시스템을 위한 새로운 메모리 페이지 교체 정책을 제안하였다. 제안한 정책에서는 상변화메모리가 쓰기 연산에 취약하다는 점에 착안하여 스토리지에 발생하는 쓰기량을 줄이기 위해 수정된 페이지가 메모리로부터 방출되는 것을 최대한 유예하는 방식을 채택하였다. 한편, 메모리가 부족한 상황에서는 이러한 수정 페이지의 방출 유예가 페이지 폴트를 증가시킬 우려가 있으므로 제안한 정책에서는 부하 상태를 고려해서 메모리에 여유 공간이 있는 경우에 한해 수정 페이지의 방출을 유예하도록 하였다. 다양한 메모리 참조 트레이스를 재현하는 실험을 통해 제안한 정책이 메모리 시스템의 성능을 저하시키지 않으면서 스토리지 쓰기량을 크게 절감함을 확인하였다. 본 논문의 결과는 메모리 요구량이 많은 머신러닝용 시스템에 적극 활용될 수 있을 것으로 기대된다[15, 16].

References

[1] S. Ng, "Advances in disk technology: performance issues," IEEE Computer, vol. 31, no. 5, pp. 75-81, 1998. DOI: https://doi.org/10.1109/2.675641

[2] T. Kim and H. Bahn, "Implementation of the storage manager for an IPTV set-top box," IEEE Transactions on Consumer Electronics, vol. 54, no. 4, pp. 1770-1775, 2008. DOI: https://doi.org/10.1109/TCE.2008.4711233

[3] R. Karedla, J.S. Love, and B.G. Wherry, "Caching strategies to improve disk system performance," IEEE Computer, vol. 27, no. 3, pp. 38-46, 1994. DOI: https://doi.org/10.1109/2.268884

[4] O. Kwon, H. Bahn, and K. Koh, "Popularity and prefix aware interval caching for multimedia streaming servers," Proc. IEEE Conf. on Computer and Information Technology, pp. 555-560, 2008. DOI: https://doi.org/10.1109/CIT.2008.4594735

[5] I. Shin, "Performance evaluation of applying shallow write in SSDs with internal cache," The Journal of KIIT, vol. 17, no. 1, pp. 31-38, 2019. DOI: https://doi.org/10.14801/jkiit.2019.17.1.31

[6] 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

[7] 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

[8] 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 Transactions on Computers, vol. 63, no. 9, pp. 2187-2200, 2014. DOI: https://doi.org/10.1109/TC.2013.98

[9] S. Yoo, Y. Jo, and H. Bahn, "Integrated scheduling of real-time and interactive tasks for configurable industrial systems," IEEE Transactions on Industrial Informatics, vol. 18, no. 1, pp. 631-641, 2022. DOI: https://doi.org/10.1109/TII.2021.3067714

[10] 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

[11] 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

[12] S. Lee and H. Bahn, "Characterization of Android memory references and implication to hybrid memory management," IEEE Access, vol. 9, pp. 60997-61009, 2021. DOI: https://doi.org/10.1109/ACCESS.2021.3074179

[13] H. Yoon, K. Cho, and H. Bahn, "Storage type and hot partition aware page reclamation for NVM swap in smartphones," Electronics, vol. 11, no. 3, 2022. DOI: https://doi.org/10.3390/electronics11030386

[14] N. Nethercote and J. Seward, "Valgrind: a framework for heavyweight dynamic binary instrumentation," ACM SIGPLAN Notices, vol. 42, no. 6, pp. 89-100, 2007. https://doi.org/10.1145/1273442.1250746

[15] N. Choi, J. Oh, J. Ahn, and K. Kim, "A development of defeat prediction model using machine learning in polyurethane foaming process for automotive seat," Journal of the Korea Academia-Industrial cooperation Society, vol. 22, no. 6, pp. 36-42, 2021. DOI: https://doi.org/10.5762/KAIS.2021.22.6.36

[16] J. Jeong, H. Kim, and J. Chun, "Automatic augmentation technique of an autoencoder-based numerical training data," The Journal of The Institute of Internet, Broadcasting and Communication, vol. 22, no. 5, pp. 75-86, 2022. DOI: https://doi.org/10.7236/JIIBC.2022.22.5.75

저 자 소 개

반 효 경(정회원)

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

박 윤 주(정회원)

- 2015년 2월 : 이화여자대학교 컴퓨터공학과 학사
- 2015년 3월 ~ : 이화여자대학교 컴퓨터공학과 통합과정
- 주관심분야: 운영체제, 스토리지 시스템, 임베디드 시스템

※ This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C1009275) and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.RS-2022-00155966, Artificial Intelligence Convergence Innovation Human Resources Development (Ewha Womans University)).