

https://doi.org/10.7236/JIIBC.2022.22.2.77  
JIIBC 2022-2-11

## 고속 스토리지 탑재에 따른 데스크탑과 스마트폰 환경의 페이지 크기 영향력 분석

### Impact Analysis for Page Size of Desktop and Smartphone Environments under Fast Storage Media

박윤주\*, 반효경\*\*

Yunjoo Park\*, Hyokyung Bahn\*\*

**요 약** 최근 고속 스토리지의 출현으로 메모리 시스템의 관리 단위인 페이지의 설정에 대한 재고가 필요한 시점에 이르렀다. 본 논문에서는 고속 스토리지의 탑재에 따라 페이지의 크기가 메모리 성능에 어떠한 영향을 미치는지를 분석하였다. 구체적으로는 데스크탑과 스마트폰 환경에서 페이지 크기에 따른 워크로드별 TLB 적중률 및 페이지 부재율을 분석하였다. 본 논문의 분석 결과 데스크탑 시스템에서는 페이지 크기의 영향력이 시스템 및 워크로드 상황에 의존적임을 확인하였다. 반면, 스마트폰 시스템의 경우 고속 스토리지가 탑재되더라도 페이지 크기가 메모리 성능에 미치는 영향이 크지 않으며, 워크로드에 따라서는 민감하지 않은 것을 확인하였다. 본 논문의 분석은 고속 스토리지 환경에서 페이지 크기를 시스템 및 워크로드 환경에 맞게 최적 설정하는 데에 도움을 줄 수 있을 것으로 기대된다.

**Abstract** Due to the recent advent of fast storage media, the memory management system needs to reconsider the configuring of a page unit. In this paper, we analyze the effect of the page size on memory performance as fast storage is adopted. Specifically, we analyze the TLB hit ratio and the page fault ratio as the workload and the page size are varied in desktop and smartphone environments. Our analysis shows that the influence of the page size depends on the system and workload conditions in desktop systems. However, in smartphone systems, the effect of the page size on memory performance is not large, and is not also sensitive to workloads. We expect that the analysis of this paper will be helpful in configuring the page size of given workloads under the system with fast storage media.

**Key Words** : fast storage, page size, memory performance, desktop, smartphone

## 1. 서 론

전통적인 스토리지인 하드 디스크는 메모리 매체인 DRAM에 비해 수만배 이상의 접근시간을 필요로 한다<sup>1)</sup>.<sup>2, 3)</sup> 따라서, 한번의 디스크 접근시 가급적 많은 양의 데

이터를 전송하도록 설계하고, 한번 읽어온 데이터는 메모리에 캐싱을 통해 재사용시 추가적인 디스크의 접근을 최대한 억제하는 방법을 취한다<sup>3, 4)</sup>. 캐싱이 효과적인 것은 대부분의 시스템에서 인기있는 데이터와 그렇지 않은 데이터를 접근 기록을 통해 구분하는 것이 가능하기 때

\*정회원, 이화여자대학교 컴퓨터공학과

\*\*정회원, 이화여자대학교 컴퓨터공학과

접수일자 2022년 3월 10일, 수정완료 2022년 3월 30일

게재확정일자 2022년 4월 8일

Received: 10 March, 2022 / Revised: 30 March, 2022 /

Accepted: 8 April, 2022

Corresponding Author: bahn@ewha.ac.kr

Dept. of Computer Engineering, Ewha University, Korea

문이다<sup>14, 5, 6</sup>.

한편, 최근 플래시메모리가 널리 사용되면서 스토리지와 메모리의 접근 속도 차이가 크게 줄어들고, NVM(Non-volatile Memory) 등 일부 고속 스토리지의 경우 물리적인 성능이 메모리와 유사한 수준에 이르게 되었다<sup>17, 8</sup>. 이에 따라 한번의 스토리지 접근시 많은 양의 데이터를 읽어 오는 것보다 꼭 필요한 데이터만을 읽어오는 것이 메모리 공간 부족시 더 효율적인 상황이 되었다. 이는 하드 디스크와 달리 NVM과 같은 유형의 스토리지에서는 데이터의 전송량이 많으면 이에 비례해서 스토리지 접근시간이 늘어나는 특성을 가지기 때문이다.

통상적으로 메모리는 페이지 단위로 관리되며, 요청 데이터가 메모리에 존재하지 않을 경우 스토리지에서 읽어오는 단위로 페이지이다. 페이지가 메모리에 존재하지 않는 상황을 페이지 부재(page fault)라고 부르며, 하드 디스크와 같은 느린 스토리지 환경에서는 효율적인 캐싱을 통해 페이지 부재율을 줄이는 것이 메모리 접근시간을 개선하는 가장 주요한 방향이다. 그러나, 고속 스토리지 환경에서는 페이지 부재율의 개선이 메모리 접근시간 향상으로 직결된다고 보기 어렵다. 하드 디스크 환경에서는 페이지 크기와 무관하게 스토리지 접근시간이 거의 동일했으나, NVM 스토리지 하에서는 페이지 크기를 증가시킬 경우 스토리지 접근시간이 늘어나므로 페이지 부재율이 개선되더라도 한번의 페이지 부재시 더 오랜 시간이 소요되는 문제점이 있기 때문이다. 또한, 고속 스토리지 환경에서는 스토리지 접근시간이 빨라져 메모리 주소변환시간의 영향력이 상대적으로 커진 것도 페이지 부재율이 절대적이던 하드 디스크 환경과는 차별화된 상황이 되었다.

본 논문에서는 고속 스토리지의 탑재에 따라 페이지 크기가 메모리 접근시간에 미치는 영향을 주소변환 측면과 페이지 부재율 측면에서 분석한다. 특히, 본 논문에서는 데스크탑 환경과 스마트폰 환경에서 고속 스토리지 탑재시 페이지 크기가 메모리 성능에 미치는 영향을 각각 분석한다. 본 논문의 분석 결과 데스크탑 환경에서는 페이지 크기가 메모리 성능에 미치는 영향이 워크로드에 매우 의존적이지만, 스마트폰의 경우 워크로드에 민감하지 않고 유사한 영향을 미치는 것을 확인하였다. 본 논문의 분석은 고속 스토리지 환경에서 페이지 크기를 시스템 및 워크로드 환경에 맞게 최적 설정하는 데에 도움을 줄 수 있을 것으로 기대된다.

## II. NVM 스토리지와 페이지 크기

본 장에서는 NVM이 고속 스토리지로 사용되는 환경에서 페이지 크기가 메모리 성능에 미치는 영향을 분석한다. 표 1은 이러한 분석을 위해 본 논문에서 사용한 메모리 참조 트레이스의 특성을 보여주고 있다. 본 논문에서는 5종의 데스크탑 워크로드에 대해 페이지 크기 변화에 따른 트레이스 재현 실험을 통해 고속 스토리지 환경에서 페이지 크기의 영향을 분석하였다.

메모리 접근 시간은 크게 주소변환시간과 데이터 접근 시간으로 구성된다. 메모리를 접근하기 위한 기계어는 메모리의 위치를 프로그램에 의존적인 논리적 주소로 표현한다. 따라서, 실제 메모리 위치를 접근하려면 논리적 주소를 물리적 주소로 변환해야 하며, 이러한 주소변환은 메인메모리에 존재하는 페이지테이블 또는 고속의 주소변환 캐쉬인 TLB(Translation Lookaside Buffer)를 통해 이루어진다. TLB를 통한 주소변환이 더 빠르지만, TLB는 엔트리의 수가 한정되어 모든 주소변환이 TLB를 통해 이루어질 수는 없으며, 따라서 주소변환을 빠르게 하기 위해서는 TLB를 통한 주소변환 비율인 TLB 적중률을 높이는 것이 중요하다.

주소변환이 끝나면 변환된 주소를 가지고 물리적 메모리의 데이터 접근이 이루어진다. 이때, 해당 데이터가 메모리에 올라와 있지 않은 경우 스토리지를 통해 해당 데이터를 메모리로 먼저 적재해야 하며, 이 상황을 페이지 부재라고 부른다. 전통적인 스토리지 환경에서는 메모리 접근시간을 빠르게 하기 위해 페이지 부재율을 줄이는 것이 가장 중요한 미션이었다. 그러나, 고속 스토리지 환경에서는 메모리와 스토리지 사이의 성능 격차가 크게 줄어들면서 페이지 부재율뿐 아니라 TLB 적중률을 높이는 것도 메모리 접근시간 개선에 중요한 요소로 자리매김하게 되었다.

표 1. 데스크탑 워크로드 트레이스의 특성

Table 1. Characteristics of desktop workload.

Workload	Memory access counts		
	Write	Read	Total
freecell	60040	430135	490175
gedit	132822	1600941	1733763
gqview	645399	265286	610685
kghostview	103540	1442595	1546135
xmms	978242	190697	1168939

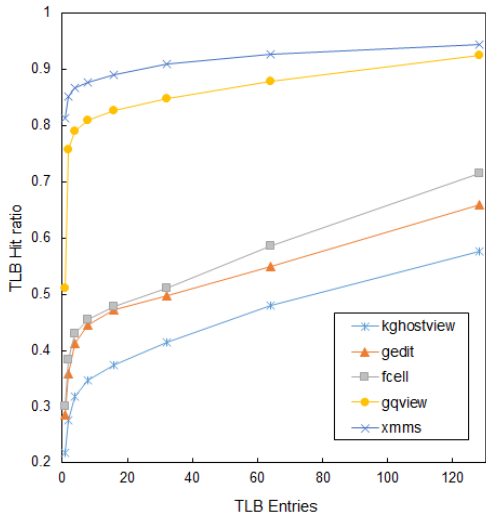


그림 1. TLB 엔트리 수에 따른 TLB 적중률  
 Fig. 1. TLB hit ratio as a function of TLB entries.

그림 1은 TLB의 엔트리 수가 변함에 따라 5종의 워크로드에 대한 TLB 적중률을 보여주고 있다. 그림에서 보는 것처럼 TLB 엔트리의 수가 증가함에 따라 TLB 적중률이 개선되는 것을 확인할 수 있다. 다만 워크로드의 종류에 따라 이러한 TLB 적중률의 추세는 상이하게 나타났다. xmms와 gqview의 경우 엔트리의 수가 20개 이하일 때 TLB 적중률이 급격하게 개선되지만 이후에는 엔트리 수 증가에 따른 성능 개선 효과는 그다지 크지 않았다. 반면, freecell, gedit, kghostview의 경우에는 TLB 적중률의 수치가 상대적으로 낮으며, 엔트리 수가 100개를 넘어갈 때까지 지속적인 성능 개선이 나타나는 것을 확인할 수 있었다. 이러한 차이는 워크로드마다 메모리 참조의 특성이 다르기 때문으로 분석할 수 있다. 소수의 인기 있는 페이지만 집중적으로 참조되는 워크로드의 경우 적은 수의 엔트리로도 주소변환정보를 TLB에 대부분 담을 수 있으므로 엔트리 수가 일정 수준 이상에서는 성능 개선 효과가 낮아지는 특성을 나타낸다. 반면, 다수의 페이지들이 골고루 참조되는 워크로드에서는 TLB 적중률이 상대적으로 낮을 수밖에 없고 엔트리 수가 늘어남에 따라 성능 개선도 꾸준히 나타나는 특성을 보인다.

그림 2는 페이지 크기가 변함에 따른 5종 워크로드의 TLB 적중률을 보여주고 있다. 그림에서 보는 것처럼 페이지 크기가 증가함에 따라 TLB 적중률이 개선되는 것을 확인할 수 있다. 이는 페이지 크기가 커질수록 한정된

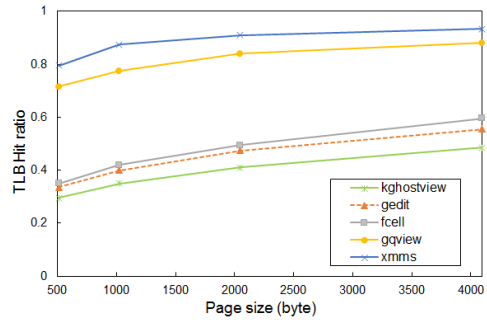


그림 2. 페이지 크기에 따른 TLB 적중률  
 Fig. 2. TLB hit ratio as a function of page size.

TLB 캐쉬가 더 넓은 메모리 영역의 주소변환을 커버할 수 있기 때문이다. 이와 같은 TLB 적중률의 개선은 페이지 테이블을 접근하는 빈도를 줄여 궁극적으로 메모리 주소변환 시간을 개선하는 데에 기여하게 된다.

한편, 그림 1에서와 마찬가지로 페이지 크기에 따른 TLB 적중률 역시 워크로드에 따라 차이가 나타나는 것을 확인할 수 있다. gqview와 xmms의 경우 모든 페이지 크기에 대해 TLB 적중률이 상대적으로 높으며, 페이지 크기가 1KB 이상인 구간부터는 페이지 크기 증가에 따른 개선 효과가 미미한 것을 확인할 수 있다. 반면, freecell, gedit, kghostview에서는 TLB 적중률이 페이지 크기 증가에 따라 지속적으로 개선되는 것을 확인할 수 있다. 이를 통해 TLB 적중률을 높이기 위해서는 페이지 크기를 증가시키는 것이 효과적임을 알 수 있으나, 그 민감도는 워크로드에 따라 다르다는 결론을 내릴 수 있다.

그림 3은 5종의 워크로드에 대해 페이지 크기가 증가함에 따른 페이지 부재율을 보여주고 있다. 그림에서 보는 것처럼 페이지 부재율은 워크로드에 따라 매우 다른 특성을 나타내었다. 페이지 부재율의 절대치도 워크로드에 따라 다르게 나타났으며, 페이지 크기 변화에 따른 추세 또한 워크로드에 의존적이었다. freecell과 gqview의 경우 페이지 크기가 4KB에 이를 때까지 성능 개선이 지속적으로 나타난 반면, xmms는 페이지 크기가 2KB 이하일 때에는 가파른 성능 개선이 나타나고 그 이상의 페이지 크기에서는 완만한 성능 개선을 나타내었다. gedit의 경우 페이지 크기가 1KB를 넘어가면서 오히려 페이지 부재율이 나빠지는 결과를 나타내었다. 워크로드에 따른 이러한 차이는 각 워크로드별 메모리 사용량 및 페이지 간의 인기 편향성이 얼마나 강한지에 따라 다르게 나타나기 때문으로 분석할 수 있다.

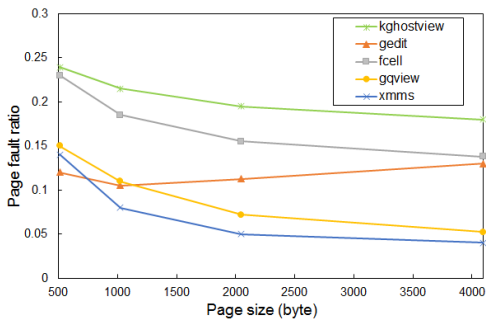


그림 3. 페이지 크기에 따른 페이지 부재율  
Fig. 3. Page fault ratio as a function of page size.

요약하자면 데스크탑 시스템의 경우 페이지 크기에 따른 TLB 적중률 및 페이지 부재율의 성향이 워크로드에 따라 다르게 나타나므로 최적의 페이지 크기 또한 시스템 및 워크로드 상황에 의존적임을 의미한다고 할 수 있다.

### III. 스마트폰 환경에서의 페이지 크기

최근 스마트폰에서 다양한 앱이 실행되면서 워크로드의 특성에 맞는 시스템의 관리가 점점 중요해지고 있다<sup>[9]</sup>. 본 장에서는 스마트폰 시스템에서 페이지 크기에 따른 메모리 접근시간의 영향력을 분석한다. 본 장에서의 실험에는 표 2에 표시된 5종의 안드로이드 워크로드에 대한 메모리 참조 트레이스를 사용하였다.

그림 4는 5종의 워크로드에 대해 TLB 엔트리 수가 변함에 따른 TLB 적중률을 보여주고 있다. 데스크탑 워크로드에서와 마찬가지로 TLB 적중률은 엔트리 수가 증가함에 따라 꾸준히 개선되는 것을 확인할 수 있다. 단, 데스크탑의 경우와 달리 5종의 워크로드 모두 유사한 수치를 나타내었으며, TLB 엔트리의 수가 20개 이하인 경우에서 급격한 성능 개선을 나타내고 그 이후부터는 점진적인 개선을 나타내었다.

그림 5는 페이지 크기가 변함에 따른 TLB 적중률을 보여주고 있다. 이 실험 역시 데스크탑의 경우와 마찬가지로 페이지 크기 증가에 따라 TLB 적중률이 개선되는 결과를 확인할 수 있었다. 이는 고정된 수의 TLB 엔트리로 더 많은 메모리 용량에 대한 주소변환 정보를 커버할 수 있기 때문이다. 그러나, 스마트폰 워크로드의 경우 모든 페이지 크기 구간에서 TLB 적중률 수치가 매우 높기 때문에 일정 한계를 넘어서는 페이지 크기는 주소 변환 성능에 그다지 영향을 미치지 않음을 알 수 있다.

표 2. 스마트폰 워크로드 트레이스의 특성  
Table 2. Characteristics of smartphone workload.

Workload	Memory access counts		
	Write	Read	Total
facebook	2045716	11607339	13653055
angrybirds	3822479	14368068	18201717
youtube	3162229	15034275	18196504
farmstory	2101818	13122852	15224670
chrome	4104436	16895563	20999999

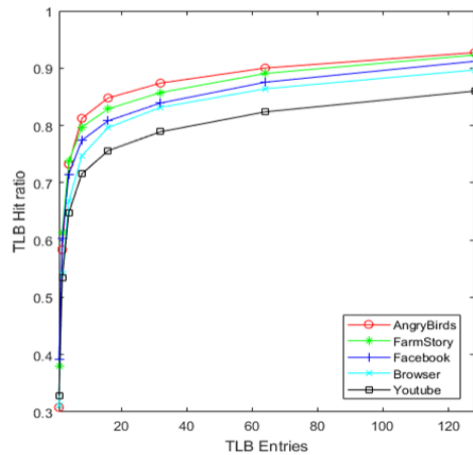


그림 4. TLB 엔트리 수에 따른 스마트폰 TLB 적중률  
Fig. 4. TLB hit ratio of smartphones as a function of TLB entries.

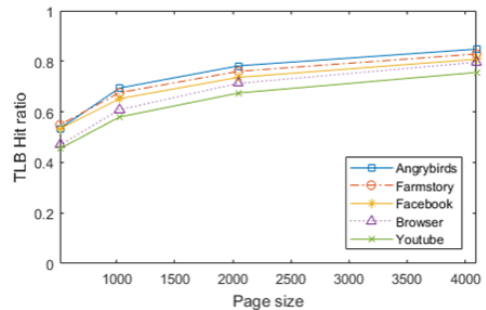


그림 5. 페이지 크기에 따른 스마트폰 TLB 적중률  
Fig. 5. TLB hit ratio of smartphones as a function of page size.

그림 6은 페이지 크기가 변함에 따른 모바일 워크로드의 페이지 부재율을 보여주고 있다. 그림에서 보는 것처럼 모든 워크로드에서 페이지 크기가 증가함에 따라 페이지 부재율이 감소하는 것을 확인할 수 있다. 데스크탑 시스템의 경우 워크로드의 종류에 따라 페이지 부재율의

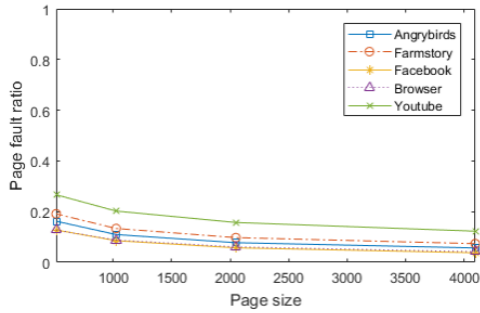


그림 6. 페이지 크기에 따른 스마트폰 페이지 부재율  
 Fig. 6. Page fault ratio of smartphones as a function of page size.

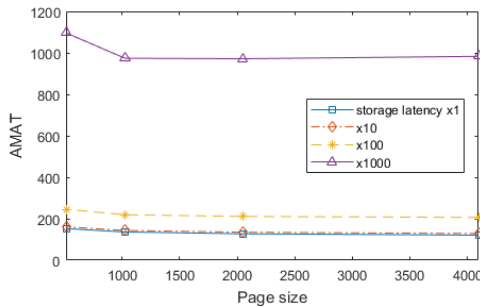


그림 7. 페이지 크기에 따른 평균 메모리 접근 시간  
 Fig. 7. Average memory access time as a function of the page size.

추세가 상이했던 것과 달리 스마트폰에서는 모든 워크로드에서 유사한 결과를 나타내었다. 이는 최적의 페이지 크기가 워크로드의 특성에 의존적이던 데스크탑과 달리 모바일 환경에서는 워크로드에 따른 차이가 거의 없었던 것을 의미한다.

페이지 크기 증가시 페이지 부재율이 개선되는 것은 한번의 스토리지 접근으로 이후에 사용될 데이터까지 미리 읽어오기 때문에 발생하는 효과이다. 데스크탑 워크로드에서는 페이지 크기 증가시 페이지 부재율이 오히려 저하되는 경우도 있었는데 이는 메모리 용량이 부족하여 한정된 메모리 내에 다양한 데이터를 보관하기가 힘들어지기 때문으로 볼 수 있다. 본 장의 모바일 실험에서는 어떤 워크로드의 경우에도 이러한 현상이 발생하지 않은 것을 알 수 있는데, 이는 안드로이드가 가용 메모리 공간이 부족할 경우 애플리케이션을 강제 종료시키는 LMK(Low Memory Killer)를 사용하여 메모리 부족 상황을 미연에 방지하기 때문으로 추측된다<sup>[10]</sup>.

그림 7은 페이지 크기가 변함에 따른 모바일 워크로드

의 평균 메모리 접근 시간을 보여주고 있다. 이 그래프에서는 고속 스토리지의 접근시간이 메모리와 동일한 경우부터 1000배 느린 경우까지에 대한 5종 워크로드의 평균 메모리 접근 시간을 보여주고 있다. 그림에서 보는 것처럼 스토리지가 메모리보다 1000배 느린 경우에는 페이지 크기가 1KB 미만일 때 성능이 저하되는 모습을 보였으나, 이를 제외하고는 어떠한 경우에도 페이지 크기가 메모리 성능에 민감하지 않은 결과를 나타내었다.

본 장의 내용을 요약하자면 안드로이드와 같은 모바일 환경에서는 고속 스토리지가 탑재되더라도 페이지 크기가 메모리 성능에 미치는 영향력이 데스크탑에서처럼 크지 않다는 것을 의미한다.

## IV. 결 론

본 논문에서는 메모리 시스템의 관리 단위인 페이지가 고속 스토리지의 탑재에 따라 어떠한 영향을 미치는지를 분석하였다. 데스크탑 시스템의 경우 페이지 크기에 따른 TLB 적중률 및 페이지 부재율의 추세가 워크로드에 따라 상이하게 나타나는 것을 확인하였으며, 이에 따라 최적의 페이지 크기 또한 시스템 및 워크로드 상황에 맞게 설정할 필요성이 있음을 확인하였다. 이에 비해 스마트폰 시스템의 경우 고속 스토리지가 탑재되더라도 페이지 크기가 메모리 성능에 미치는 영향이 크지 않았으며, 워크로드에 따른 민감성도 그다지 높지 않은 것을 확인하였다. 본 논문의 분석은 고속 스토리지가 활성화되는 다양한 차세대 컴퓨팅 환경에서 시스템 및 워크로드 환경에 맞게 페이지 크기를 최적 설정하는 데에 도움을 줄 수 있을 것으로 기대된다.

## 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 Trans. 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 Conference on Computer and Information Technology, pp. 555-560, 2008.  
DOI: <https://doi.org/10.1109/CIT.2008.4594735>
- [5] E. Lee, J. Whang, U. Oh, K. Koh, and H. Bahn, "Popular channel concentration schemes for efficient channel navigation in internet protocol televisions," IEEE Trans. Consumer Electronics, vol. 55, no. 4, pp. 1945-1949, 2009.  
DOI: <https://doi.org/10.1109/TCE.2009.5373754>
- [6] 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: [https://doi.org/10.1016/S0140-3664\(01\)00365-6](https://doi.org/10.1016/S0140-3664(01)00365-6)
- [7] Intel Optane Technology,  
<https://www.intel.com/content/www/us/en/architecture-and-technology/intel-optane-technology.html>
- [8] 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>
- [9] 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>
- [10] Low Memory Killer Daemon,  
<https://source.android.com/devices/tech/perf/lmkd>

## 저 자 소 개

### 박 윤 주(정회원)



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

### 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, 임베디드시스템

※ This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C1009275) and the ICT R&D program of MSIT/IITP (2018-0-00549, extremely scalable order preserving OS for manycore and non-volatile memory).