

<https://doi.org/10.7236/JIIBC.2020.20.1.149>  
JIIBC 2020-1-21

# 이기종 메모리로 구성된 스마트폰 메모리의 페이지 배치 기법

## A Page Placement Scheme of Smartphone Memory with Hybrid Memory

이소윤\*, 반효경\*\*

Soyoon Lee\*, Hyokyung Bahn\*\*

**요약** 본 논문은 스마트폰 시스템에서 DRAM과 NVRAM으로 구성된 이기종 메모리를 위한 페이지 배치 기법을 제안한다. 이기종 메모리에 관한 기존 연구와 달리 본 논문은 메모리 접근에 대한 오프라인 분석에 기반하여 메모리 페이지를 배치한다. 이는 스마트폰 메모리 접근이 애플리케이션의 종류와 무관하게 특정 주소 영역에 집중적으로 나타나며, 쓰기 연산에 있어 그 편향성이 일관되게 나타난다는 점을 반영한 것이다. 제안한 기법은 오프라인 분석 결과를 토대로 NVRAM에 쓰기 트래픽이 적게 발생하도록 페이지 배치를 수행하며, 실험 결과 NVRAM에 발생하는 쓰기량을 성능 저하 없이 평균 61% 줄이는 것을 확인할 수 있었다.

**Abstract** This paper presents a new page placement policy for DRAM/NVRAM hybrid main memory in smartphones. Unlike previous studies on hybrid memory systems, this paper performs the placement of pages based on the offline analysis of memory access behaviors as smartphone's memory accesses are skewed to a certain address ranges, which is consistent regardless of smartphone applications, specially for write operations. Thus, we aim at reducing the write traffic to NVRAM by the offline analysis results. Experimental results show that the proposed policy reduces the write traffic to NVRAM by 61% on average without performance degradations.

**Key Words** : Smartphone, NVRAM, DRAM, hybrid memory, page placement, application

### 1. 서 론

스마트폰의 배터리 소모에서 메모리가 차지하는 비중이 높은 것으로 보고되고 있다<sup>[1][5]</sup>. 이는 메모리로 사용되는 DRAM이 휘발성 매체이므로 읽기 쓰기 연산이 없더라도 지속적인 전력 재공급(refresh)이 필요하기 때문이다<sup>[6]</sup>. 최근, 이러한 메모리의 전력소모를 줄이기 위해 차

세대메모리(NVRAM)가 주목받고 있다<sup>[7][8]</sup>. NVRAM은 비휘발성 매체이므로 DRAM과 달리 전력 재공급 연산이 필요하지 않아 배터리 소모를 줄일 수 있으나, 쓰기 연산이 느리거나 쓰기 연산의 전력 소모가 큰 약점이 있다. 이를 보완하기 위해 NVRAM에 소량의 DRAM을 같이 사용하는 하이브리드 메모리 아키텍처가 제안된 바 있다<sup>[2]</sup>. 하이브리드 메모리 아키텍처에서는 DRAM과

\*비회원, 이화여자대학교 컴퓨터공학과

\*\*정회원, 이화여자대학교 컴퓨터공학과(교신저자)

접수일자: 2019년 11월 18일, 수정완료: 2020년 1월 8일

게재확정일자: 2020년 2월 7일

Received: 18 November, 2019 / Revised: 8 January, 2020 /

Accepted: 7 February, 2020

Corresponding Author: bahn@ewha.ac.kr

Dept. of Computer Engineering, Ewha University, Korea

NVRAM이 동일한 메모리 주소 공간을 사용하며 DRAM이 쓰기 연산을 최대한 흡수하여 NVRAM의 쓰기 취약성을 완충하는 것이 중요하다.

본 논문은 하이브리드 메모리 아키텍처에서 소량의 DRAM이 메모리 쓰기를 최대한 흡수하도록 하는 페이지 배치 기법을 제안한다. 제안하는 기법은 스마트폰 애플리케이션의 메모리 쓰기 참조가 특정 주소 영역에 집중되어 있다는 점을 분석하고 이러한 페이지들을 DRAM에 배치한다. 이를 위해 각 앱별 쓰기 참조 분포를 오프라인으로 분석한 후, 이에 기반한 페이지 순위를 부여하고 순위가 높은 페이지를 DRAM에 배치한다. 기존의 기법들이 페이지 자체의 배치를 온라인으로 결정하여 많은 오버헤드를 발생시킨 것과 달리 본 연구는 스마트폰의 페이지 참조 특성을 오프라인으로 분석하는 것이 효율적임을 보인다.

최근 한 연구에 따르면 온라인 분석에 기반한 데이터 배치에서 잘못된 예측으로 인한 불필요한 페이지 이동이 전체 실행시간의 평균 25%까지 증가시켰다고 보고하고 있다<sup>[3]</sup>. 이에 비해 본 논문은 오프라인으로 페이지 순위를 미리 조사한 후 이에 기반하여 데이터를 배치한다는 점에서 차별화된다. 한편, DRAM에 배치할 페이지의 인기 순위는 오프라인으로 결정하여 불필요한 오버헤드를 줄이지만, DRAM에 배치할 페이지의 수는 시스템 상태에 맞게 조절하여 효율성을 높인다.

본 논문에서는 DRAM의 페이지 폴트율을 주기적으로 모니터링 한 후, 오프라인으로 분석한 페이지의 순위 중 몇 위까지를 DRAM에 배치할지를 페이지 폴트율에 따라 동적으로 조절한다. 즉, 페이지 폴트율이 높은 경우 DRAM 배치 페이지의 순위를 축소하고 페이지 폴트율이 낮은 경우 DRAM 배치 페이지의 순위를 확대한다. 이는 페이지 폴트율에 의한 성능저하를 방지하면서 NVRAM 쓰기량을 줄이는 효과를 얻게 한다.

본 논문의 구성은 다음과 같다. II장에서는 스마트폰 앱의 쓰기 참조 특성을 분석한다. III장에서는 제안하는

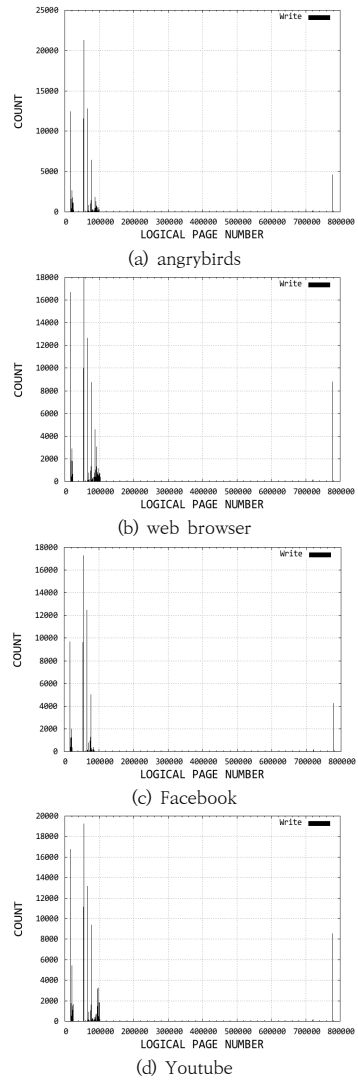


그림 1. 주소 영역에 따른 메모리 쓰기 연산 분포  
Fig. 1. Distribution of memory writes for address spaces.

표 1. 실험에 사용한 트레이스 추출 정보

Table 1. Trace extraction information used in the experiments.

Workload	Memory footprint (KB)	Memory footprint by writes (KB)	Ratio of operations (reads : writes)	Memory access count			
				Total	Instruction read	Data read	Data write
angrybirds	76,936	45,724	3.72 : 1	18,003,944	521,326	13,780,318	3,702,300
web browser	84,736	53,856	3.73 : 1	17,929,520	1,154,807	13,223,286	3,551,427
facebook	44,284	20,608	6.90 : 1	10,159,776	368,566	8,551,807	1,239,403
youtube	68,640	40,948	4.66 : 1	17,986,993	983,231	14,000,933	3,002,829

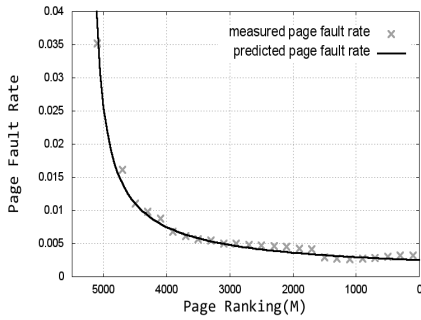


그림 2. 실측 페이지 폴트율과 모델 예측치 비교  
Fig. 2. Comparison of measured page fault rate and estimated one.

페이지 배치 기법을 설명한다. IV장에서는 제안한 기법의 우수성을 실험을 통해 검증한다. 끝으로 V장에서는 본 논문의 결론을 제시한다.

## II. 스마트폰 앱의 쓰기 참조 특성 분석

본 논문은 스마트폰 앱들의 수행시 발생하는 메모리 참조 트레이스를 추출하고 쓰기 참조가 어떤 특징을 나타내는지를 분석한다. 트레이스 추출은 Valgrind 3.8.1의 cachegrind tool set을 이용하였다. 캐시메모리를 통해 흡수되지 않고 실제 메모리에 전달되는 요청들을 추출하기 위해 본 논문에서는 cachegrind에 캐시 레이아웃(L1, L2, LLC)에 대한 애플레이터를 추가하였고 안드로이드 기반 넥서스 폰을 이용하여 실제 메모리 참조 트레이스를 추출하였다. 표 1은 추출된 트레이스의 특성을 보여주고 있다.

그림 1은 각 앱들의 메모리 주소 영역별로 발생하는 쓰기 참조의 분포를 보여주고 있다. 그림에서 볼 수 있듯이 쓰기 참조는 전체 메모리 주소 영역에 골고루 분포하여 발생하지 않고 특정 영역에 집중적으로 발생함을 알 수 있다. 특히, shared library와 linker, stack이 위치한 영역에 쓰기 참조가 주로 발생하는 것으로 분석되었다. 또한 참조 카운트의 차이는 있으나 이러한 현상은 앱의 종류에 무관하게 나타나는 것으로 확인되었으며, 이러한 오프라인 분석을 토대로 페이지 배치시 주소를 통해 쓰기 연산이 빈번한 페이지를 DRAM에 배치하는 것이 가능함을 의미한다.

## III. 페이지 배치 기법

II장의 분석 결과를 바탕으로 쓰기 참조 빈도가 높은 페이지를 DRAM에 배치하여 메모리 쓰기를 최대한 DRAM이 흡수하는 페이지 배치 기법을 제안한다. 제안하는 기법은 각 페이지의 쓰기 참조 순위를 오프라인으로 조사하고 이에 기반하여 DRAM에 배치할 페이지를 결정한다. 쓰기 참조 발생 시, 요청된 페이지의 참조 순위가 임계랭킹 M보다 상위에 랭크된 페이지이면 DRAM에 배치하고, 그렇지 않으면 NVRAM에 배치하며, 임계랭킹은 시스템의 메모리 상황에 따라 동적으로 조절한다.

메모리 공간 부족으로 인해 DRAM에 배치된 페이지가 방출되는 경우 본 논문은 이를 스토리지로 내려 보낸다. 한편, DRAM의 크기가 한정적이므로 이로 인해 페이지 폴트율이 증가하는 것을 막기 위해 본 논문은 주기적으로 DRAM의 페이지 폴트율을 조사하고 이에 근거하여 임계랭킹 M을 조절한다.

임계랭킹 M을 작게 설정하면 쓰기 연산이 매우 잦은 페이지만을 DRAM에 배치하게 되며, 이럴 경우 NVRAM에 배치되는 페이지들에서도 상당 부분의 쓰기 연산이 발생하게 된다. 그러나, DRAM에 배치되는 페이지의 수가 적기 때문에 DRAM에서 발생하는 페이지 폴트 또한 줄어들게 된다. 반대로, 임계랭킹 M을 크게 설정하는 경우 DRAM으로 배치될 수 있는 페이지가 많아져 쓰기 연산의 상당 부분을 DRAM이 흡수할 수 있는 반면, DRAM에서의 페이지 폴트율은 증가하게 된다.

메모리 시스템의 목표는 느린 스토리지 접근을 줄이는 것이므로 페이지 폴트율 개선이 가장 중요한 척도라 할 수 있다. 따라서, 시스템 내에서 DRAM의 페이지 폴트율이 높은 경우 제안하는 기법은 임계랭킹 M을 감소시켜 DRAM에 배치될 수 있는 페이지 수를 줄여 페이지 폴트 상황을 줄인다. 반면, 메모리 작업집합이 작아 페이지 폴트율이 낮은 상황에서는 더 많은 페이지들을 DRAM에 배치하기 위해 임계랭킹 M을 증가시킨다. 이는 DRAM이 더 많은 쓰기 연산들을 흡수하도록 하여 NVRAM의 쓰기 발생량을 감소시키게 된다.

한편, 페이지 폴트율이 높은지 낮은지의 판단을 위해 본 논문은 절대적 수치에 의존하지 않고 현재의 임계랭킹 M에서 발생하는 페이지 폴트율과 현재의 임계랭킹보다 단위 크기만큼 낮은 임계랭킹(M-Δm)에서 발생하게 될 페이지 폴트율을 비교한다. 이는 DRAM의 크기가 일정수준 커졌을 때 페이지 폴트율이 어떤 영향을 받는지 조사하는 것으로, 현재 메모리 배치가 DRAM에 오버

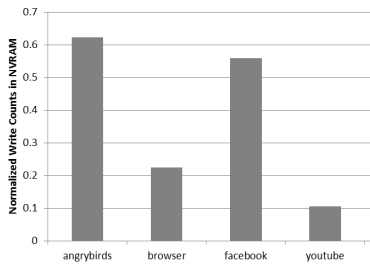


그림 3. 앱별 NVRAM 쓰기 발생량  
Fig. 3. Write count of NVRAM for each application.

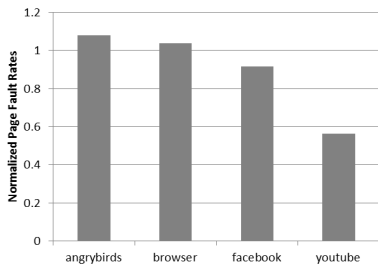


그림 4. 앱별 페이지 폴트율  
Fig. 4. Page fault rate for each application.

헤드를 초래하고 있는 상황인지를 확인할 수 있다.

임계랭킹의 동적 조절을 위해서는 임계랭킹이  $M$ 인 현 시점의 페이지 폴트율 외에도 임계랭킹이  $M+\Delta m$ 인 경우와  $M-\Delta m$ 인 경우의 페이지 폴트율을 예측할 수 있어야 한다. 이를 위해 본 논문은 Belady's lifetime function을 이용하여 페이지 폴트율을 예측한다<sup>[4]</sup>. Belady's lifetime function은 메모리 크기에 따른 페이지 폴트율을 예측할 수 있는 함수로 알려져 있으며, 본 논문에서는 임계랭킹에 따른 페이지 폴트율 예측을 위해 이를 모델링하고 실제값과 예측값 사이에 큰 오차없이 잘 적용됨을 확인하였다. 그림 2는 Belady's lifetime function을 이용한 예측 모델과 실제 측정된 페이지 폴트율을 비교한 그래프로 예측 모델이 실측값에 상당히 근사함을 확인할 수 있다.

#### IV. 성능 평가

본 장에서는 제안하는 기법의 효과를 시뮬레이션 실험을 통해 검증한다. 실험에서 전체 메모리 크기는 각 앱별 총 메모리 사용량의 90%가 되도록 설정하였으며 DRAM

의 크기는 전체 메모리 크기의 20%가 되도록 설정하였다. 그림 3은 쓰기 연산이 발생한 모든 페이지를 DRAM에 배치하는 기존의 방법<sup>[2]</sup>과 본 논문에서 제안한 기법의 NVRAM 쓰기 발생량을 비교한 것이다. 그림에서 y축은 제안한 기법이 NVRAM에 발생시킨 쓰기량을 기존 기법이 발생시킨 쓰기량을 기준으로 정규화해서 표시한 것이다. 그림에서 보는 것처럼 제안하는 기법은 NVRAM 쓰기 발생량을 크게 감소시켰으며, 구체적으로는 38~88%까지 개선되는 것을 확인할 수 있었다.

그림 4는 모든 쓰기 발생 페이지를 DRAM에 배치하는 기존 기법 대비 제안하는 기법의 페이지 폴트율이 어느 정도인지를 보여주고 있다. 그림에서 보는 것처럼 angrybirds, web browser 등에서는 제안하는 기법이 약간의 페이지 폴트율 증가를 나타낸 것을 확인할 수 있다. 이는 페이지를 DRAM에서 방출할 때 제안하는 기법은 NVRAM이 아닌 스토리지로 직접 방출하기 때문에 페이지 폴트율에서 약간의 손해가 발생할 수 있기 때문이다. 그러나, 본 논문에서는 페이지 폴트율로 인한 전체 시스템 성능 저하가 커지지 않도록 주기적으로 페이지 폴트율을 모니터링하고 페이지 폴트율이 높아지는 경우 한시적으로 쓰기 참조 페이지들을 NVRAM에 좀 더 배치하여 페이지 폴트율을 조절하게 된다. 이에 따라 NVRAM 쓰기 발생량의 감소 정도에 비해 합리적인 수준의 페이지 폴트율을 나타내는 것을 확인할 수 있다. 또한, facebook, youtube의 경우 오히려 페이지 폴트율이 기존 기법보다도 개선된 결과를 나타내었다. 이는 쓰기 참조의 특성을 데이터 배치에 반영함으로써 좀더 효율적으로 메모리 사용을 가능하게 한 효과라고 볼 수 있다.

#### V. 결 론

본 논문은 NVRAM과 DRAM으로 구성된 하이브리드 메모리 아키텍처에서 NVRAM의 쓰기 취약성을 완충하기 위한 페이지 배치 기법을 제안하였다. 제안한 기법은 스마트폰의 메모리 쓰기 참조가 특정 주소 영역에 편중되어 나타난다는 점을 오프라인 분석으로 규명하고, 분석 결과를 토대로 쓰기 참조가 빈번한 페이지를 DRAM에 배치한다. 실제 앱에서 추출한 메모리 참조 트레이스를 재현하는 시뮬레이션 실험을 통해 제안하는 기법이 기존 기법 대비 페이지 폴트율에 큰 차이가 없으면서 NVRAM의 쓰기 발생량을 평균 61%, 최대 88%까지 줄일 수 있음을 확인하였다.

## References

- [1] S. Nam, K. Cho, and H. Bahn, "Tight Evaluation of Real-Time Task Schedulability for Processor's DVS and Nonvolatile Memory Allocation," *Micromachines*, Vol. 10, No. 6, 2019.  
DOI: <https://doi.org/10.3390/mi10060371>
- [2] S. Lee, H. Bahn, S. 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>
- [3] S. Bock, B. Childers, R. Melhem, and D. Mosse, "Concurrent page migration for mobile systems with OS-managed hybrid memory," In *Proc. Computing Frontiers*, pp.1-10, 2014.
- [4] E. Lee, K. Koh, and H. Bahn, "Dynamic memory allocation for real-time and interactive jobs in mobile devices," *Electronics Letters*, Vol. 46, No. 6, pp.401-403, 2010.  
DOI: <https://doi.org/10.1049/el.2010.3367>
- [5] S. Nam, 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/JIIBC.2017.17.2.1>
- [6] M. Cho, K. Han, and W. Jang, "A Pre-Refresh Technique for Improving DRAM Performance," *The Journal of KIIT*, Vol.16, No.10, pp.1027-1035, 2018.  
DOI: <http://doi.org/10.14801/jkiit.2018.16.10.27>
- [7] E. Lee, H. Bahn, and S. Noh, "Unioning of the Buffer Cache and Journaling Layers with Non-volatile Memory," In *Proc. 11th USENIX FAST Conf.*, pp. 73-80, 2013.
- [8] E. Lee, J. Kim, H. Bahn, S. Lee, S. Noh, "Reducing Write Amplification of Flash Storage through Cooperative Data Management with NVM," *ACM Transactions on Storage*, Vol. 13, No. 2, 2017.  
DOI: <https://doi.org/10.1145/3060146>

## 저 자 소 개

### 이 소 윤(비회원)



- 2004년 2월 : 이화여자대학교 컴퓨터공학과 학사
- 2006년 2월 : 이화여자대학교 컴퓨터공학과 석사
- 2011년 2월 : 이화여자대학교 컴퓨터공학과 박사
- 2011년 3월 ~ : 이화여자대학교 컴

퓨터공학과 연구교수.

- 주관심분야 : 운영체제, 스토리지 시스템

### 반 호 경(정회원)



- 1997년 2월 : 서울대학교 계산통계학과 학사
- 1999년 2월 : 서울대학교 전산학과 석사
- 2002년 2월 : 서울대학교 컴퓨터공학부 박사.
- 2002년 9월 ~ : 이화여자대학교 컴

퓨터공학과 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).