The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 16, No. 4, pp.15-20, Aug. 31, 2016. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2016.16.4.15
JIIBC 2016-4-3

# 플래시 스토리지의 성능 지연 방지를 위한 비휘발성램 기반 쓰기 증폭 감소 기법

# NVM-based Write Amplification Reduction to Avoid Performance Fluctuation of Flash Storage

이은지*, 정민성**, 반효경***

**Eunji Lee*, Minseong Jeong**, Hyokyung Bahn*****

**요 약** 플래시 메모리는 초소형 전자기기부터 미디어 서버에 이르기까지 현대의 다양한 시스템에서 스토리지로 활용되고 있다. 플래시 메모리의 쓰기 증폭 (Write Amplification)은 가비지 컬렉션에서 발생하는 것으로 불규칙적인 성능의 주요 원인으로 지적되고 있다. 갑작스러운 속도지연은 실시간성 미디어를 위한 스토리지 시스템에서 치명적인 단점이 될 수 있다. 본 논문은 비휘발성램을 플래시 메모리 스토리지의 버퍼캐시로 사용하고 두 계층 간의 협동적 데이터 관리를 통해 플래시 메모리의 쓰깆 WAF를 절감하는 기법에 대해 제안한다. 비휘발성램에 캐쉬된 데이터는 플래시 메모리에서 가비지 컬렉션 수행 시 복사하지 않도록 한다. 이것은 복사되는 페이지의 수를 감소시켜 스토리지의 성능 및 내구성을 향상시킨다. 제안된 기법은 ssdsim 시뮬레이터에 구현되었으며 WAF와 응답시간의 표준편차를 각각 51.4%와 35.4% 개선할 수 있음을 보인다.

**Abstract** Write amplification is a critical factor that limits the stable performance of flash-based storage systems. To reduce write amplification, this paper presents a new technique that cooperatively manages data in flash storage and nonvolatile memory (NVM). Our scheme basically considers NVM as the cache of flash storage, but allows the original data in flash storage to be invalidated if there is a cached copy in NVM, which can temporarily serve as the original data. This scheme eliminates the copy-out operation for a substantial number of cached data, thereby enhancing garbage collection efficiency. Experimental results show that the proposed scheme reduces the copy-out overhead of garbage collection by 51.4% and decreases the standard deviation of response time by 35.4% on average.

**Key Words :** Non-volatile Memory, Flash Memory, Buffer Cache, Storage System

## Ⅰ. 서 론

최근 PCM(Phase Change Memory), STT-MRAM (Spin torque transfer magnetic RAM), 3D XPoint 등과 같은 비휘발성램 기술 발전이 가속화 됨에 따라 1-2년 내에 상용화가 될 것으로 전망되고 있다[1]. 바이트 단위의 접근이 가능하면서도 내구성을 제공하는 비휘발성 메모리는, 기존 DRAM 기반의 컴퓨터 스토리지 계층구조에서 다양한 형태로 사용되어 성능 및 내구성을 향상시키는 데 기여할 수 있을 것으로 기대되고 있다[2]. 본 논문

*정회원, 충북대학교 소프트웨어학과
**준회원, 충북대학교 소프트웨어학과
***정회원, 이화여자대학교 컴퓨터공학과(교신저자)
접수일자 : 2016년 6월 11일, 수정완료 : 2016년 8월 2일
게재확정일자 : 2016년 8월 5일

Received: 11 June, 2016 / Revised: 2 August, 2016 / Accepted: 5 August, 2016
***Corresponding Author: bahn@ewha.ac.kr
Dept. of Computer Engineering, Ewha University, Korea

에서는 NVM(Non-volatile Memory)을 플래시 스토리지의 캐쉬장치로 사용할 때 두 계층 간의 협동적인 데이터 관리를 통해 플래시 메모리의 쓰기 증폭현상을 감소시키는 방법을 제안하고자 한다.

플래시 메모리는 임베디드 장치에서 데이터 센터에 이르기까지 다양한 시스템에서 광범위하게 사용된다. 그러나 플래시 메모리는 덮어쓰기를 허용하지 않는 물리적 특성 때문에 주기적인 데이터 복사로 인해 쓰기 증폭 현상(Write Amplification)이 나타난다. 구체적으로, 한 번 기록된 데이터를 제자리에서 변경하는 것이 불가능하고 새로운 위치에 데이터를 적은 후 이전 데이터는 무효화 시킨다. 이 때 새로운 데이터를 기록할 공간은 쓰기 전 삭제(erase before write) 를 통해 초기화 되어야 하는데 삭제 연산의 단위(블록, 통상 64개의 페이지)가 쓰기 연산의 단위(페이지, 통상 4KB) 와 달리, 삭제 연산 시 삭제 대상 블록에 무효화되지 않고 남아있는 페이지들은 새로운 빈 블록에 복사해야 한다. 이러한 불필요한 데이터 복사로 인한 쓰기 증폭 현상(Write Amplification)은 쓰기 속도가 느리고 최대 쓰기 횟수에 제한이 있는 플래시 메모리에서는 성능 및 내구성 저하의 주요 원인으로 지적되고 있다[3, 4].

본 논문은 플래시 메모리의 쓰기 증폭을 완화시키기 위해 비휘발성램 버퍼캐시에 캐쉬 된 데이터를 플래시 메모리의 삭제 연산 시 새로운 블록에 복사하지 않고 지울 수 있도록 허용하는 기법을 제안한다. 비휘발성램은 전원이 없이도 데이터를 영속적으로 저장할 수 있기 때문에 향후 데이터를 복구할 수 있다. 이를 위해 버퍼캐시는 비휘발성램에 저장된 캐쉬 데이터를 일시적인 데이터가 아니라 데이터 원본으로 관리하여 기존 시스템과 동일한 수준의 내구성과 일관성을 제공하도록 한다. 비휘발성 버퍼캐시와 플래시 메모리 간의 협동적인 데이터 관리는 PCIe 등의 최신 스토리지 인터페이스에서 구현 가능한 기법으로 신뢰성 저하 없이 스토리지 시스템의 성능과 내구성을 향상시킬 수 있다[5-8].

제안하는 기법은 Microsoft Research 에서 개발한 DiskSim의 SSD 확장버전인 SSDsim에 구현하였다[9]. 다양한 워크로드 기반 실험에서 플래시 메모리의 쓰기 증폭 지수(Write Amplification Factor, WAF)를 평균 51.4% 감소시켰으며 그로 인해 응답 시간이 평균 15% 감소되었고, 쓰기 증폭 현상 결과에 따라 응답 시간의 표준편차는 35.4% 감소하였다.

본 논문은 다음과 같이 구성된다. Ⅱ장에서는 다양한 워크로드에서 SSD의 쓰기 증폭 현상에 대해 조사한다. Ⅲ장에서는 제안하는 협동적 관리 기법의 알고리즘에 대해 자세히 기술하고, Ⅳ장에서는 제안하는 기법의 성능 평가 및 장단점을 분석하고 Ⅴ장에서는 결론을 맺는다.

## Ⅱ. 워크로드별 SSD의 WAF 분석

WAF(Write Amplification Factor)란 플래시 메모리가 물리적으로 덮어쓰기를 허용하지 않기 때문에 데이터를 변경하기 위해서는 반드시 원래의 데이터를 지우고 다시 쓸 때 발생하는 엄청난 수의 불필요한 쓰기 이다. 상용 SSD의 내부 정보는 사용자에게 제공되지 않는 이유로 우리는 높은 정확도를 가지는 SSD 시뮬레이터인 SSDsim을 사용하여 SSD의 WAF를 조사하였다(실험의 환경설정에 대한 자세한 설명은 4장에서 할 예정임). 우리는 2개의 합성 워크로드(Random과 Sequential) 과 2개의 실제 워크로드(JEDEC과 OLTP)를 사용하여 WAF를 측정하였다. 합성 워크로드는 SSDsim에서 내부 워크로드 생성기를 사용하여 500만개의 random 패턴 쓰기 명령과 sequential 패턴 쓰기 명령을 생성하였다. 각 명령은 8 섹터 (4KB) 단위와 총 footprint 가 20GB이다. 표 1은 실험에서 사용된 실제 워크로드들의 특성들을 보여준다. JEDEC (JEDEC 219A)는 SSD 내구성 검사를 위해 사용된 워크로드이고 OLTP 트레이스는 금융 트랜잭션들을 위한 I/O 접근들을 생성한다[10, 11].

측정을 시작하기 전에 자유 블록의 수를 전체 플래시 블록의 5% 미만이 될 때까지 순차적으로 데이터를 씀으로써 시뮬레이터를 준비 시켰다. 결과들은 각 워크로드마다 10번씩 반복 실행하여 측정하였다. 그림 1에서 보는 바와 같이 Random 쓰기의 경우 WAF가 평균 2.84로 나타났고 Sequential 쓰기는 WAF가 1.0을 유지하였다. WAF의 경우 플래시 메모리에서 GC 수행 시 삭제 대상 블록에 유효 페이지 수가 많을수록 수치가 증가하는데 워크로드가 Random 패턴을 가질수록 하나의 블록에 있는 페이지들이 무효화되는 시점이 달라 WAF가 증가하게 된다. Sequential 패턴의 경우 하나의 블록에 쓰여진 페이지들은 순차적으로 함께 무효화되기 때문에 별도의 유효 페이지 복사 없이 삭제가 가능하다.

**표 1. 사용한 워크로드 특성**
**Table 1. Summary of workload characteristics.**

| | JEDEC | OLTP |
|---|---|---|
| # of ops. | 5,000,000 | 9,034,179 |
| Ratio of ops. | write 89%, trim 6%, flush 5% | read 52%, write 48% |
| Footprint. | 31.2GB | 30.7GB |

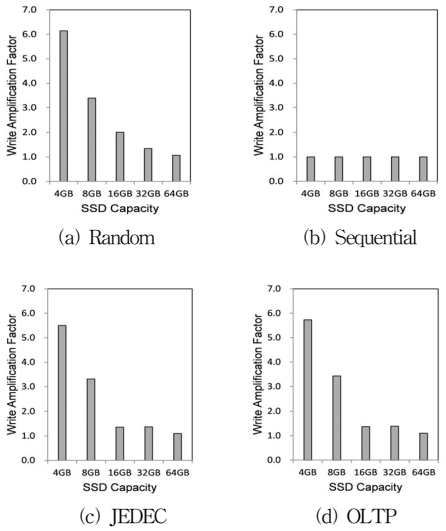



**그림 1. 다양한 워크로드들의 WAF 분석**
**Fig. 1. Write amplification factor for various workloads.**

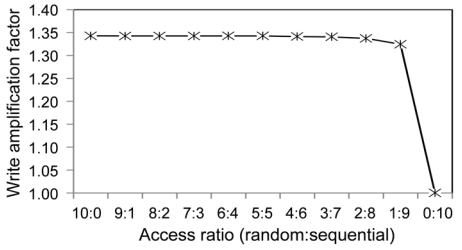



**그림 2. Random 과 Sequential 비율에 의한 WAF**
**Fig. 2. Write amplification factor as access ratio is varied.**

실제 워크로드인 경우 대부분 Random 과 Sequential 접근이 혼합된 형태로 나타내기 때문에 우리는 Random 과 Sequential 접근이 혼합된 워크로드를 발생시켜 각 비율에 따른 WAF 변화를 조사해보았다. 그림 2는 SSD가 32GB 일 때 Random과 Sequential 접근 비율 변화에 따른 WAF를 보여준다. 매우 흥미로운 결과는 상당부분 Sequential 이고 Random 패턴이 작은 비율로 혼합되더라도 모두 Random 패턴을 지닌 워크로드와 비슷한 수준으로 WAF가 증가한다는 것이다. 이것은 실제 워크로드에서는 대부분 Random 워크로드에서 관찰된 수준으로 WAF가 증가함을 의미한다.

이러한 분석은 실제 I/O 워크로드의 WAF 조사결과와도 일치하는데 JEDEC(그림 1. c)와 OLTP(그림 1. d)에서 WAF는 Random 워크로드(그림 1. a)와 유사하게 나타났다. 즉 플래시 메모리를 스토리지로 사용하는 경우 요청된 쓰기보다 평균 2.07배 많은 쓰기가 플래시 내부에서 발생하고 용량이 작은 경우에는 쓰기량이 5.7배까지 증가할 수 있다는 것이다. 이러한 쓰기 증폭 현상은 플래시 메모리의 응답시간 지연을 심화시키고 셀의 노화를 가속화시켜 스토리지 수명에도 심각한 악영향을 줄 수 있다.

이에 우리는 비휘발성램이 메인메모리로 사용될 때, 버퍼캐시와 플래시 메모리의 협동적인 데이터 관리를 통해 플래시 메모리의 WAF를 감소시키는 기법에 대해 제안한다.

## Ⅲ. 협동적 데이터 관리 기법

### 1. 동작 방식

기존 캐싱 시스템은 DRAM과 같은 휘발성 매체를 사용했기 때문에 버퍼캐시 내에 다른 복사본이 있더라도 스토리지에 있는 원본 데이터는 반드시 보존 되어야 한다. 그러나 비휘발성 버퍼캐시에서는 데이터가 캐쉬와 스토리지에서 모두 영속적으로 저장될 수 있다. 따라서 둘 중 하나의 데이터는 필요한 경우 제거할 수 있다. CDM(Cooperative Data Management) 기법은 이러한 점에 착안하여 캐쉬에 데이터가 존재한다면 그들의 유효한 데이터를 복사하는 과정 없이 해당 블록을 재활용 할 수 있도록 한다.

그림 3은 NVM 버퍼캐시와 플래시가 협동할 때 버퍼캐시와 플래쉬 데이터의 상태 다이어그램이다. 스토리지와 캐쉬에 존재하는 데이터는 페이지 단위로 관리 된다. 스토리지 데이터의 상태는 “valid”, “invalid”, “removable”로 이루어지고 캐쉬된 데이터는 clean, dirty 상태를 가질 수 있다. S0 상태는 플래시에 데이터가 저장되어 있는 초기 상태를 나타낸다. 사용자 요청에 의해 스토리지의 데이터가 캐쉬에 저장되면 S1 상태로 변한다. 이때 데이터

는 캐쉬와 플래시 둘 다 존재하기 때문에 스토리지에 있는 데이터는 필요시 삭제 가능한 은 “removable” 상태가 된다. 만약 캐쉬된 데이터가 수정되거나 스토리지 원본이 삭제되는 경우 버퍼캐시에 있는 데이터만이 가장 최신의 유효한 데이터로 남아있기 때문에 스토리지, 스토리지 데이터의 경우 존재한다면 “invalid” 상태가 된다. 마지막으로 캐쉬 된 dirty 상태인 데이터가 캐쉬에서 쫓겨난다면 플래시로 write-back 되고 스토리지에만 원본 데이터가 존재하는 S0 상태가 된다. clean 상태(S1)의 캐쉬 된 데이터가 교체된다면 write-back 없이 캐쉬에서 무효화시킨다.

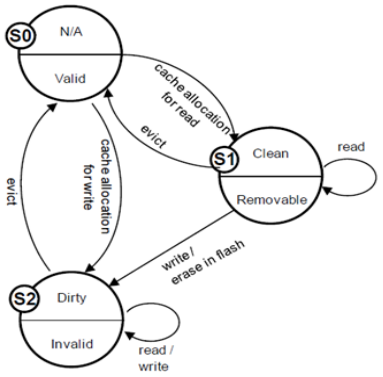



**그림 3. 비휘발성 캐쉬와 플래시 스토리지가 협동할 때 캐쉬와 스토리지 데이터의 상태**
**Fig. 3. Statics of cache and storage data when the NVM cache and flash storage cooperate.**

### 2. 적용 아키텍쳐

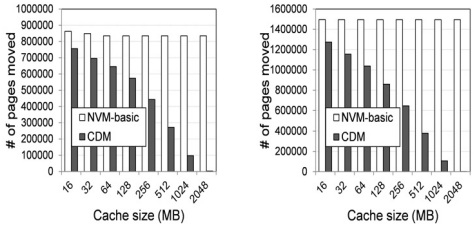
제안된 기법은 크게 두 가지의 아키텍쳐에서 적용 가능하다. 첫 번째는 호스트 페이지 캐시를 비휘발성램으로 사용할 때 스토리지와 호스트 캐쉬 간에 협동적인 데이터 관리를 하는 것이다. 이 경우 상대적으로 대용량의 비휘발성램을 탑재할 수 있기 때문에 WAF 감소효과가 클 것으로 예상되나 호스트와 디바이스 간의 커뮤니케이션 오버헤드가 상당히 발생할 수 있다. 호스트 OS와 스토리지 시스템은 캐쉬와 스토리지 시스템에서 일어나는 상태의 변화를 실시간으로 동기화해야 하기 때문이다. 이를 위해서는 호스트 인터페이스의 확장 뿐 아니라 파일시스템의 관리 매커니즘의 재구현도 필요로 한다[12]. 최근 NVMe나 Universal PCI Express 같은 스토리지 인터페이스는 사용자 특성에 맞게 인터페이스를 확장하는 것이 가능해지고 있지만[13], 빈번한 정보 교류는 상당한 성능저하로 이어질 것이다. 본 연구는 향후 이러한 오버헤드를 최소화할 수 있는 방안을 고안하고자 한다.

좀 더 현실적인 아키텍쳐는 플래시 스토리지 내부 버퍼로 NVM을 사용하고, 해당 버퍼와 플래시  사이에서 데이터를 협동적으로 관리하는 것이다. 이러한 구조에서는 FTL(Flash Translation Layer)에서 비휘발성램과 스토리지 내부의 존재하는 데이터의 상태를 일괄적으로 관리하기 때문에 캐쉬와 스토리지 간의 데이터 공유로 인한 성능저하가 발생하지 않는다. 또한 수정이 필요한 계층도 FTL로 단일화가 되기 때문에 개발 용이성 측면에서도 좋다. 이에 본 논문에서는 후자의 아키텍쳐를 타겟으로 성능평가를 진행하였다.

## Ⅳ. 성능 평가

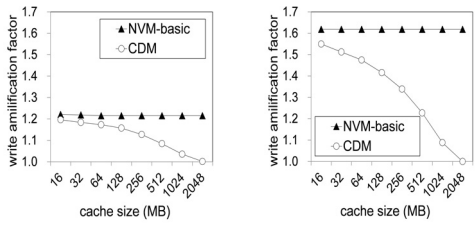
제안된 협동적 데이터 관리 기법은 Microsoft Research에서 개발한 DiskSim의 SSD 확장 버전인 SSDsim 시뮬레이터에 구현되었다. SSDsim 은 SLC NAND 플래시 메모리를 에뮬레이트 해주는 플래시 시뮬레이터로 디폴트 환경설정 값으로 실험을 진행하였다. 우리는 SSDsim 시뮬레이터에 비휘발성 캐쉬 모듈을 추가하고 FTL 내부에 CDM 기법을 구현하였다. 비휘발성램 캐쉬는 LRU 교체 정책을 사용하며 4KB 페이지 단위로 관리된다. 플래시와 캐쉬 내의 데이터의 상태정보를 저장하기 위하여 FTL 페이지 테이블 엔트리에 상태를 나타낼 수 있는 비트정보를 추가하였다. 또한 가비지 컬렉션 모듈을 수정하여 “removable” 상태의 페이지는 copy-out 을 하지 않고 무효화 시키도록 하였다. 우리는 협동적 관리기법(CDM)을 비휘발성램을 기존 DRAM 과 같은 일시적인 캐쉬로 사용하는 방식(NVM-basic)과 비교하며 성능을 측정하였다.

그림 4는 캐쉬 크기가 변할 때 가비지 컬렉션(GC) 수행 동안 발생한 복사된 페이지 수를 그래프로 나타낸 것이다. 그림에서 보는 바와 같이 제안된 기법은 기존 캐싱 방식 대비 복사되는 페이지의 수를 크게 감소시켰다. 구체적으로, CDM은 NVM-basic 대비 복사되는 페이지의 수를 JEDEC과 OLTP에서 각각 48.3% 와 54.4% 줄였다. 특히 캐쉬 크기가 증가함에 따라 감소되는 복사 페이지의 수는 더욱 증가한다. 비휘발성램의 용량이 클수록 삭제가능하거나 미리 무효화 된 플래시 데이터가 증가할 수 있기 때문이다.



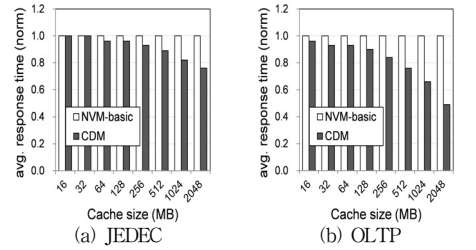

(a) JEDEC (b) OLTP

**그림 4. 복사된 페이지의 수**
Fig. 4. Number of pages copied out.



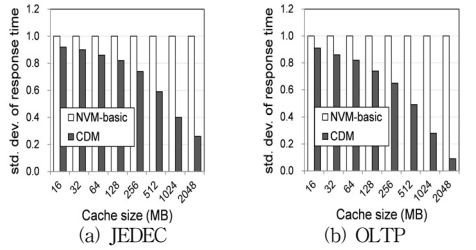

(a) JEDEC (b) OLTP

**그림 5. 쓰기 증폭 요소**
Fig. 5. Write amplification factor.




(a) JEDEC (b) OLTP

**그림 6. 평균 응답 시간**
Fig. 6. Average response time.




(a) JEDEC (b) OLTP

**그림 7. 응답 시간의 표준 편차**
Fig. 7. Standard deviation of response time.

그림 5는 버퍼캐시 사이즈에 따른 WAF를 보여준다. 제안된 기법은 기존 방식 대비 WAF를 JEDEC과 OLTP 워크로드에서 각각 2.1-17.6% 와 4.3-38.2% 감소시켰다. 이것은 GC 발생 시 복사되는 페이지의 수를 상당히 절감한 결과이며, 쓰기량의 감소는 플래시 메모리의 성능과 내구성 향상에 크게 기여할 수 있다.

그림 6과 7은 CDM과 NVM-basic의 평균 응답 시간과 응답 시간의 표준편차를 보여준다. 그래프에서 보여주는 바와 같이, CDM은 기존 방식 대비 JEDEC과 OLTP에서 각각 평균 응답 시간은 9.7%와 20.3%, 평균 표준 편차는 31%와 39% 를 감소시켰다. 특히 표준편차의 감소는 플래시 메모리의 갑작스러운 성능저하로 사용자의 불편함을 초래하고 있는 서버 환경에서 QoS(Quality of Service)를 제공하는 데에 크게 기여할 것으로 기대된다.

## V. 결 론

본 논문은 NVM이 캐쉬에 적용될 때 플래시 스토리지를 위한 새로운 데이터 관리 기법을 제안했다. 제안된 기법은 비휘발성 캐쉬와 플래시 스토리지에서 데이터를 협동적으로 관리함으로써 GC(Garbage Collection)의 효율성을 향상시키고 전체적인 성능 및 편차를 개선한다. 구체적으로 데이터가 비휘발성 캐쉬에서 영속성을 보장받는 경우 플래시 내부에서는 해당 데이터를 다른 곳으로 복사(copy-out) 없이 블록을 삭제할 수 있도록 하는 것이다. 제안된 기법은 SSDsim 을 통해 성능을 평가하였으며, 실험결과 GC의 copy-out 페이지의 수를 평균 51.4% 감소시켰다.

## References

[1] https://en.wikipedia.org/wiki/3D_XPoint
[2] E. Lee, S. Yoo, H. Bahn, “Performance Evaluation and Analysis of NVM Storage for Ultra-Light Internet of Things,” The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 15, No. 6, pp. 181-186, 2015
[3] Y. Lu, J. Shu, and W. Zheng, “Extending the Lifetime of Flash-based Storage through Reducing Write Amplification from File Systems,”

Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST), pp. 73-80, 2013.

[4] P. Desnoyers, "Analytic modeling of SSD write performance," Proceedings of the 5th ACM International Systems and Storage Conference (SYSTOR), 2012.

[5] M. Yang, Y. Chang, C. Tsao, and P. Huang, "New ERA: new efficient reliability-aware wear leveling for endurance enhancement of flash storage devices," Proceedings of the 50th Annual Design Automation Conference (DAC), 2013.

[6] D. Apalkov, A. Khvalkovskiy, S. Watts, V. Nikitin, X. Tang, D. Lottis, K. Moon, X. Luo, E. Chen, A. Ong, A. Driskill-Smith, and M. Krounbi, "Spin-transfer torque magnetic random access memory (STT-MRAM)," ACM Journal on Emerging Technologies in Computing Systems, 9(2), 2013.

[7] O. Zilberberg, S. Weiss, and S. Toledo, "Phase-change memory: An architectural perspective," ACM Computing Surveys, 45(3), 2013.

[8] Y. Li and K. N. Quader, "NAND Flash memory: challenges and opportunities," Computer, pp. 23-29, 2013.

[9] N. Agrawal, V. Prabhakaran, T. Wobber, J. Davis, M. Manasse, and R. Panigrahy, "Design tradeoffs for SSD performance," Proc. USENIX ATC, pp. 57-70, 2008.

[10] JEDEC, Master trace for 128 GB SSD, http://www.jedec.org/standards-documents/docs/jesd219a_mt.

[11] UMASS trace repository, http://traces.cs.umass.edu.

[12] F. Shu, "Data set management commands proposal for ATA8-ACS2," T13 Technical Committee, United States: At Attachment:e07154r1, 2007.

[13] A. Huffman, "NVM Express: Going Mainstream and What's Next", Intel Developers Forum, 2014.

## 저자 소개

**이 은 지(정회원)**

- 2005년 2월 : 이화여자대학교 컴퓨터공학과 학사
- 2012년 2월 : 서울대학교 컴퓨터공학부 박사
- 2014년 3월 ~ : 충북대학교 소프트웨어학과 조교수

<주관심분야 : 운영체제, 스토리지, 가상화, 분산 시스템>

**정 민 성(준회원)**

- 2016년 2월 : 충북대학교 소프트웨어학과 학사
- 2016년 3월 ~ : 충북대학교 소프트웨어학과 석사과정

<주관심분야 : 운영체제, 임베디드 시스템, 가상화, 파일시스템>

**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, 스토리지 시스템, 임베디드 시스템>

※ 이 논문은 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.NRF-2014R1A1A3053505)