



## NVM 스토리지의 최적 활용을 위한 워크로드별 I/O 특성 분석

Analyzing Workload I/O Characteristics for Optimized Adoption of NVM Storage

---

저자 (Authors)	김지선, 반효경 Jisun Kim, Hyokyung Bahn
출처 (Source)	<a href="#">한국정보과학회 학술발표논문집</a> , 2016.6, 1509-1511 (3 pages)
발행처 (Publisher)	<a href="#">한국정보과학회</a> KOREA INFORMATION SCIENCE SOCIETY
URL	<a href="http://www.dbpia.co.kr/Article/NODE07017890">http://www.dbpia.co.kr/Article/NODE07017890</a>
APA Style	김지선, 반효경 (2016). NVM 스토리지의 최적 활용을 위한 워크로드별 I/O 특성 분석. 한국정보과학회 학술발표논문집, 1509-1511.
이용정보 (Accessed)	성균관대학교 과학학술정보관 115.***.170.232 2017/09/08 14:35 (KST)

---

### 저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독 계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

### Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

# NVM 스토리지의 최적 활용을 위한 워크로드별 I/O 특성 분석

김 지 선<sup>○</sup> 반 효 경

이화여자대학교 컴퓨터공학과

js.kim@ewhain.net bahn@ewha.ac.kr

## Analyzing Workload I/O Characteristics for Optimized Adoption of NVM Storage

Jisun Kim, Hyokyung Bahn

Ewha Womans University

### 요 약

최근 PCM 등 고속의 NVM 매체가 등장하면서 이를 활용한 파일시스템, 저널링 등 스토리지 성능 개선 연구가 활발히 이루어지고 있다. 하지만, 한정된 NVM을 파일시스템, 저널링, 스왑 디바이스 등 다양한 스토리지 역할 중 어떤 역할로 활용하는 것이 전체 시스템의 성능 개선 효과가 극대화되는지에 대한 분석은 이루어지지 않고 있다. 본 논문은 이러한 질문에 답하기 위해 다양한 애플리케이션별 I/O 특성을 분석하고 이에 기반한 NVM 스토리지의 최적 활용 방안에 대해 논의한다. 분석 결과 스토리지 I/O를 집중적으로 발생시키는 파티션이 파일시스템, 스왑 파티션, 저널 파티션 중 하나로 고정되지 않고 애플리케이션 카테고리별로 다르게 나타나는 것을 확인하였다. 분석결과를 토대로 그래프 응용, DB 응용, 멀티미디어 응용의 경우 소량의 NVM을 각각 스왑 파티션, 저널 파티션, 파일시스템 파티션으로 활용하는 것이 효과적임을 보인다.

## 1. 서 론

최근 PCM(Phase-change memory), STT-MRAM(spin transfer torque RAM), 3D Xpoint 등 고성능 NVM 스토리지가 출현하면서 플래시메모리 및 하드디스크와 함께 스토리지 시스템의 구성에 이들 매체가 활용될 가능성이 높아지고 있다[1-3]. NVM 스토리지의 활용을 위한 다양한 연구가 진행되고 있지만, 한정된 NVM을 파일 I/O, 저널링 I/O, 스왑 I/O 중 어떤 I/O를 흡수하는 디바이스로 활용하는 것이 전체 시스템의 성능을 극대화할 수 있는지에 대한 분석은 이루어지지 않고 있다. 본 논문은 이러한 질문에 답하기 위해 다양한 애플리케이션별 I/O 특성을 분석하고 이에 기반한 NVM 스토리지의 최적 활용 방안에 대해 논의한다.

컴퓨터 시스템에 발생하는 스토리지 I/O는 크게 파일 I/O, 저널링 I/O, 스왑 I/O로 나누어 볼 수 있다. 본 논문에서는 다양한 애플리케이션의 I/O 특성 분석 결과 스토리지 I/O를 집중적으로 발생시키는 파티션이 파일시스템, 스왑 영역, 저널 영역 중 하나로 고정되지 않고 애플리케이션 카테고리별로 다르게 나타난다는 것을 확인하였다.

특히, OLTP와 같은 DB 응용에서는 트랜잭션 단위의 관리가 필요하므로 저널 영역에 지속적인 쓰기 연산이 발생하여 저널링 I/O가 전체 I/O의 상당 부분을 차지하는 것을 확인하였다. 반면에 gnuplot과 같이 대용량의 그래프를 점 단위로 생성하는 응용의 경우 메모리 사용량이 매우 크기 때문에 스왑 I/O가 전체 I/O의 상당 부분을 차지하는 것을 확인하였다. 이는 메모리 용량이 애플리케이션의 워킹셋을 수용하기에 충분하지 않을 경우 페이지폴트로 인한 스왑 발생이 불가피하기 때문이다. 이에

비해 웹 브라우저나 멀티미디어 플레이어와 같이 데이터 파일 접근 위주의 애플리케이션에서는 전체 I/O의 상당 부분을 파일 I/O가 차지함을 분석하였다.

이와 같은 분석 결과를 토대로 소량의 NVM이 주어졌을 때 파일시스템, 저널 디바이스, 스왑 디바이스로 사용 시 각 애플리케이션이 얻게 되는 이득이 어느 정도인지를 실험을 통해 분석하였다. 실험 결과 NVM을 고정된 용도의 스토리지 파티션으로 사용할 경우 성능 개선 폭이 크지 않으며, 애플리케이션 특성에 맞게 할당하는 것이 필요하다는 것을 확인하였다. 그래프 응용의 경우 NVM을 스왑 디바이스로 활용할 때 가장 큰 성능 개선을 나타내었으며, DB 응용의 경우 저널 디바이스로 활용할 경우 가장 좋은 성능을 나타낼 것이라는 예상과는 달리 NVM의 용량이 매우 작을 경우 파일 I/O를 흡수하도록 할당했을 때 더 좋은 성능을 내는 것을 확인할 수 있었다. 이는 DB 응용의 경우 I/O의 가장 많은 비중을 저널링 I/O가 차지하고 있지만 파일 I/O에서 접근 빈도가 매우 높은 인기 블록이 존재하기 때문으로 확인되었다. 웹브라우저나 멀티미디어 플레이어와 같은 애플리케이션에서는 NVM을 파일 I/O를 흡수하도록 활용했을 때 가장 좋은 성능을 내었다. 본 논문의 분석 결과에 따라

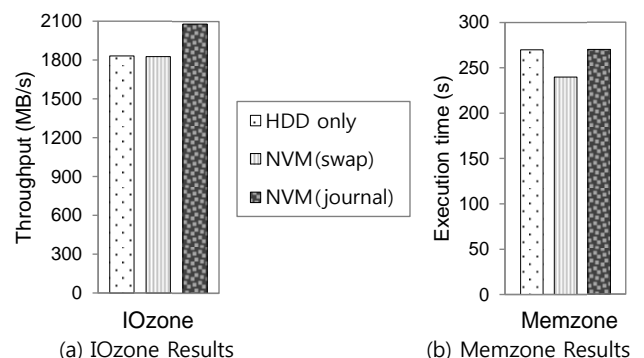


그림 1. IOzone과 Memzone의 벤치마크 결과

\* 본 연구는 산업통상자원부 및 한국산업기술평가관리원의 산업원천기술평가사업(No.10041608)과 미래창조과학부 및 한국연구재단의 이공분야기초연구지원사업(No. 2011-0028825)의 지원을 받아 수행됨.

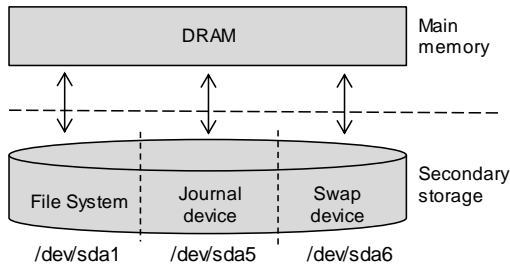


그림 2. 트레이스가 추출된 파티션 환경

NVM을 최적 배치할 경우 스토리지의 성능을 61%까지 개선할 수 있음을 확인하였다.

## 2. 연구 동기

본 장에서는 실제 시스템에서의 실측(measurement) 실험을 통해 NVM을 워크로드 특성에 맞게 활용할 필요가 있음을 보인다. 실험은 Linux kernel 3.16.0에 Ext4 파일시스템을 설치하여 수행하였다. 아직까지 NVM의 상용화가 지극히 제한적이므로 본 논문에서는 DRAM의 일부를 램디스크로 구성하여 NVM의 성능을 에뮬레이션하였다. NVM을 스왑 또는 저널 디바이스로 사용하기 위해서는 DRAM을 블록 디바이스로 인식해야 하므로 이를 위해 Ramdisk 드라이버를 적절히 활용한 NVM 디스크 드라이버를 개발하여 실험하였다. 본 논문은 HDD만 사용한 기존 시스템과 NVM을 스왑 또는 저널 디바이스로 추가한 시스템의 성능을 측정하였다. 성능 평가를 위한 벤치마크로는 IOzone과 Memzone을 사용하였다. IOzone은 I/O 요청 위주의 워크로드를 발생시키고, Memzone은 메모리 사용 위주의 워크로드를 발생시키는 벤치마크이다.

그림 1은 각각 IOzone의 처리량과 Memzone의 실행시간을 보여주고 있다. 그림 1(a)에서 보는 것처럼 IOzone의 경우 NVM을 저널 디바이스로 사용했을 때 큰 성능 개선을 나타내었다. 이는 IOzone이 많은 양의 I/O를 발생시키는 벤치마크로 느린 하드디스크 대신 빠른 NVM이 저널링 I/O를 흡수하기 때문에 좋은 성능을 나타내는 것을 의미한다. 반면, IOzone은 메모리 사용량이 많지 않으므로 NVM을 스왑 디바이스로 활용한 경우 성능 개선 효과가 거의 없는 것을 확인할 수 있다. 그림 1(b)는 Memzone의 실험결과로, 그림에서 보는 것처럼 NVM을 스왑으로 활용한 경우 상당한 성능 개선을 나타내고 있다. 실제 메모리 크기가 증가한 것은 아니지만 NVM을 통한 고속 스왑을 지원할 경우 메모리 사용량이 많은 워크로드의 성능이 확연히 개선될 수 있음을 의미한다. 이 실험은 NVM을 특정 목적의 파티션으로 고정시키는 것보다 시스템의 사용 목적에 맞게 활용하는 것이 필요하다는 것을 암시한다.

## 3. 애플리케이션 별 스토리지 I/O 특성 분석

본 장에서는 다양한 실제 애플리케이션들이 어떤 I/O를 주로 발생시키는지를 파일 I/O, 저널링 I/O, 스왑 I/O로 구분하여 트레이스를 추출하고 이를 분석한다. 각 파티션의 구분을 위해 그림 2와 같이 HDD를 파일 시스템, 저널 영역, 스왑 영역으로 나누어 실험하였다. 실험 환경은 512MB DDR3-10700 메모리와 60GB SATA HDD로 구성하였고, Ubuntu 14.04 64 bit와 Ext4 파일시스템을 사용하였다. 저널링 I/O를 별도로 추출하기 위해

표1. 스토리지 파티션 정보

Partition	Start	End	Description
/dev/sda1	2048	103999487	Filesystem
/dev/sda5	104001536	111998975	Journal device
/dev/sda6	112001024	119998463	Swap device

Ext4의 외부 저널링(external journaling) 옵션을 사용하였다. 표 1은 본 논문의 실험에 사용된 파티션별 정보를 보여주고 있다.

본 논문에서는 크게 4종류의 애플리케이션 카테고리에 대한 I/O 트레이스를 추출하였다. 이들은 그래프 생성 툴인 gnuplot, 멀티미디어 플레이어인 videoplayer, sysbench에 의해 발생하는 OLTP 워크로드, 그리고 웹 브라우저인 firefox로 구성된다. 그림 3은 각 파티션 별 I/O량을 보여 주고 있다. 그림에서 보는 것처럼 하나의 특정 파티션에서 I/O가 집중적으로 발생하지 않고, 애플리케이션 카테고리에 따라 다르게 나타나는 것을 확인할 수 있다.

그림 3(a)에서 보는 것처럼 gnuplot은 스왑 I/O가 상당히 많이 발생하였다. 그래프 응용들은 그래프의 각 점들을 계산하고 이를 화면에 표시하기 위해 많은 양의 메모리를 필요로 한다. 따라서 메모리 요구량이 애플리케이션의 워킹셋을 수용하기에 충분하지 않을 경우 많은 스왑 I/O를 발생시키게 된다. 그림 3(b)에서 보는 것처럼 OLTP는 저널링 I/O가 전체 I/O의 상당 부분을 차지하고 있다. 이는 OLTP가 일련의 트랜잭션 처리 워크로드를 가지고 있기 때문에 저널 영역에 많은 I/O를 발생시키기 때문이다. 그렇지만 OLTP는 파일시스템 영역에도 일정량 이상의 I/O가 발생하는 것을 확인할 수 있다. 이는 트랜잭션 처리 애플리케이션의 수행이 빈번한 저널링 I/O를 유발하지만 궁극적으로 파일 및 그 메타데이터에 대한 접근도 수반함을 의미한다. 그림 3(c)와 그림 3(d)에서 보는 것처럼 videoplayer와

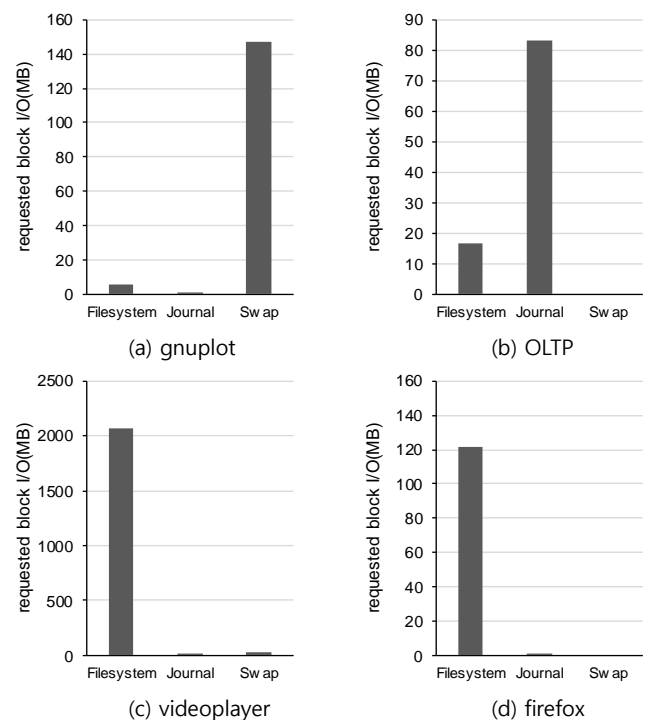


그림 3. 각 스토리지 파티션에 발생하는 I/O 요청량

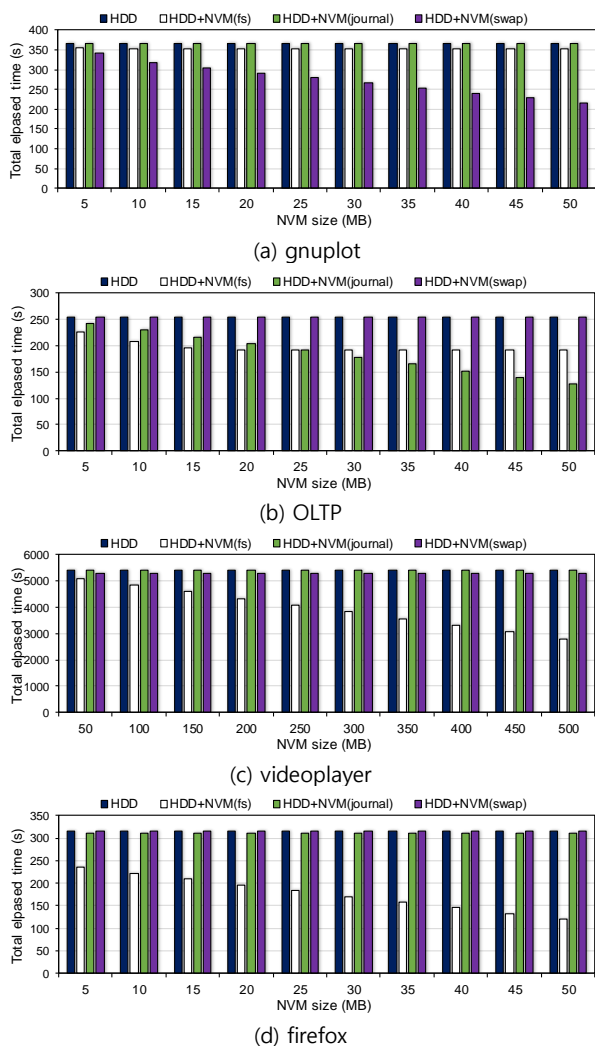


그림 4. 다양한 NVM 용량에 따른 HDD, HDD+NVM(fs), HDD+NVM(journal) 그리고 HDD+NVM(swap)의 총 I/O 실행 시간

firefox의 경우 파일 시스템 영역에서 가장 많은 I/O가 발생했다. videoplayer는 동영상 파일을 지속적으로 읽어 이를 플레이 하기 때문에 파일시스템 영역에서의 읽기 I/O가 많이 발생했으며, firefox와 같은 웹 브라우저는 웹 서버로부터 웹 객체를 읽어 이를 로컬 파일 시스템에 캐싱하고 필요시 추후 활용하기 때문에 파일 I/O가 빈번히 발생한 것을 알 수 있다.

#### 4. 워크로드에 따른 NVM 최적 할당 방안

앞 장의 분석 결과를 바탕으로 애플리케이션의 I/O 특성을 반영한 NVM의 최적 활용 방안에 대해 논의한다. 대상 아키텍처는 기존 HDD 스토리지에 소량의 NVM이 추가된 시스템이며 이러한 상황에서 NVM이 파일 I/O, 저널링 I/O, 스왑 I/O를 위한 파티션으로 사용될 경우 성능 개선이 어떻게 나타나는지를 조사하였다.

그림 4는 HDD만 사용한 시스템과 소량의 NVM을 저널 파티션, 스왑 파티션, 파일시스템 파티션으로 사용했을 때 gnuplot, OLTP, videoplayer, firefox의 실행시간을 보여주고 있다. 그림 4에서 보는 것처럼 NVM을 특정 스토리지 파티션으로 활용했을 때 일관된 성능 개선 효과를 얻을 수 없었으며, 애플리케이션 및 주어진 NVM 용량별로 최적 할당 파티션이 다르다는 것을

확인할 수 있다. 그림 4(a)에서 보는 것처럼 gnuplot의 경우 NVM을 스왑 디바이스로 사용할 때 가장 성능이 좋은 것을 확인할 수 있다. 이는 gnuplot이 많은 양의 스왑 I/O를 발생시킨 그림 3(a)와 일관성 있는 결과이다.

OLTP의 경우 그림 4(b)에서 보는 것처럼 NVM 용량에 따라 최적의 결과를 나타내는 할당 방안이 다를 수 있다. NVM 용량이 25MB보다 적을 때에는 HDD+NVM(fs)가 가장 좋은 성능을 나타내었으나, 그보다 더 큰 용량일 때에는 HDD+NVM(journal)이 가장 좋은 성능을 나타내었다. 즉, OLTP의 경우 파일 시스템에서 hot I/O를 발생시키는 블록이 존재하여 NVM 파티션 일부를 이를 흡수하기 위한 용도로 활용하는 것이 효과적이며, 이보다 더 큰 NVM이 주어진 경우 다량의 I/O를 발생시키는 저널 영역으로 할당하는 것이 적절함을 의미한다.

그림 4(c)와 그림 4(d)는 videoplayer와 firefox의 결과를 보여 주고 있다. 데이터 파일 접근이 많은 이들 애플리케이션에서는 NVM을 파일 시스템 파티션으로 할당했을 때 성능이 가장 많이 개선되는 것을 확인할 수 있다.

본 논문의 실험 결과를 통해 그래프 응용, DB 응용, 멀티미디어 응용 등이 주로 사용되는 시스템에 한정된 크기의 NVM이 주어진 경우 스왑, 저널, 파일시스템 파티션으로 각각 활용하는 것이 효과적임을 확인할 수 있었다. OLTP와 같은 DB 응용의 경우 NVM의 일부를 파일시스템의 hot I/O를 흡수하는 용도로 사용할 경우 더 큰 성능 개선을 얻을 수 있음도 확인하였다.

#### 5. 결 론

본 논문에서는 한정된 NVM 스토리지가 주어진 경우 이를 어떤 용도로 사용하는 것이 성능 개선 효과가 가장 클 것인지에 대해 다양한 애플리케이션 카테고리의 I/O 패턴 분석을 통해 살펴 보았다. 분석 결과 많은 양의 I/O가 특정 파티션에서만 발생하는 것이 아니라 애플리케이션 카테고리에 따라 다르게 나타나는 것을 확인하였다. 그래프 응용, DB 응용, 멀티미디어 응용의 경우 NVM을 각각 스왑, 저널, 파일시스템 파티션으로 활용할 경우 최대 61%의 성능 향상을 나타낼 수 있는 것으로 확인되었다. 한편, 소량의 NVM이 파일시스템 전체를 담을 수는 없으므로 본 논문의 향후 연구로는 NVM과 하드디스크를 함께 활용하여 hot I/O만을 NVM이 흡수할 수 있도록 하는 이중 매체를 위한 파일시스템 혹은 계층적 스토리지 구성에 대해 연구할 예정이다.

#### 참 고 문 헌

- [1] E. Lee et al., "On-demand Snapshot: An Efficient Versioning File System for Phase-Change Memory," IEEE Tran. Knowledge & Data Engineering, vol. 25, no. 12, pp.2841-2853, 2013.
- [2] A. Akel et al., "Onyx: a prototype phase change memory storage array," Proc. USENIX HotStorage, 2011.
- [3] A.M. Caulfield et al., "Moneta: A High-Performance Storage Array Architecture for Next-Generation, Non-volatile Memories," Proc. IEEE/ACM MICRO Conf., pp.385-395, 2010.